

**PEWA Messtechnik GmbH**

Weidenweg 21  
58239 Schwerte

Telefon: 02304-96109-0  
Fax: 02304-96109-88  
E-Mail: [info@pewa.de](mailto:info@pewa.de)



## **DPO4000 Series Digital Oscilloscopes Programmer Manual**

Copyright © Tektronix, Inc. All rights reserved.

Tektronix products are covered by U.S. and foreign patents, issued and pending. Information in this publication supersedes that in all previously published material. Specifications and price change privileges reserved.

Tektronix, Inc., P.O. Box 500, Beaverton, OR 97077

TEKTRONIX and TEK are registered trademarks of Tektronix, Inc.

### **Contacting Tektronix**

Tektronix, Inc.  
14200 SW Karl Braun Drive  
P.O. Box 500  
Beaverton, OR 97077  
USA

For product information, sales, service, and technical support:

- In North America, call 1-800-833-9200.
- Worldwide, visit [www.tektronix.com](http://www.tektronix.com) to find contacts in your area.

---

# Table of Contents

Preface .....	iii
Getting Started .....	1-1
Setting Up Remote Communications .....	1-2
Command Syntax .....	2-1
Command and Query Structure .....	2-1
Clearing the Instrument .....	2-3
Command Entry .....	2-4
Constructed Mnemonics .....	2-6
Argument Types .....	2-7
Command Groups .....	2-11
Acquisition Command Group .....	2-11
Alias Command Group .....	2-12
Calibration and Diagnostic Command Group .....	2-13
Cursor Command Group .....	2-14
Display Command Group .....	2-16
Ethernet Command Group .....	2-18
File System Command Group .....	2-18
Hard Copy Command Group .....	2-19
Horizontal Command Group .....	2-20
Mark Command Group .....	2-21
Math Command Group .....	2-22
Measurement Command Group .....	2-23
Miscellaneous Command Group .....	2-26
Save and Recall Command Group .....	2-28
Search Command Group .....	2-29
Status and Error Command Group .....	2-32
Trigger Command Group .....	2-33
Vertical Command Group .....	2-41
Waveform Transfer Command Group .....	2-43
Zoom Command Group .....	2-48
Commands Listed in Alphabetical Order .....	2-51
Status and Events .....	3-1
Registers .....	3-1
Queues .....	3-4
Event Handling Sequence .....	3-5
Synchronization Methods .....	3-7

Programming Examples ..... 4-1

Appendix A: Character Set ..... A-1

Appendix B: Reserved Words ..... B-1

Appendix C: Factory Default Setup Values..... C-1

Appendix D: GPIB Interface Specifications ..... D-1

Glossary

Index

---

# Preface

This programmer manual covers my scope



---

# Getting Started

This programmer guide provides you with the information required to use GPIB commands for remotely controlling your instrument. With this information, you can write computer programs that will perform functions such as setting the front-panel controls, taking measurements, performing statistical calculations, and exporting data for use in other programs, such as spreadsheets.

Besides the traditional GPIB electronic interface (referred to as the physical GPIB interface), your instrument is provided with a *TekVISA* GPIB-compatible interface (referred to as the virtual GPIB interface). This is a software Application Programming Interface (API) which enables you to communicate with the instrument in a variety of ways, including via the internet. With the following two exceptions, these interfaces are completely independent:

- **HEADER.** Command headers enabled or disabled on one interface are correspondingly enabled or disabled on the other interface. Refer to the command descriptions for more detailed information.
- **VERBOSE.** Verbosity enabled or disabled on one interface is correspondingly enabled or disabled on the other interface. Refer to the command description for more detailed information.

Refer to Documentation for information on related manuals and documents.

The programmer guide is divided into the following major topics:

- **Getting Started.** This topic introduces you to the online help and provides basic information about setting up your instrument for remote control.
- **Command Syntax.** This topic provides an overview of the command syntax that you will use to communicate with the instrument and other general information about commands, such as how commands and queries are constructed, how to enter commands, constructed mnemonics, and argument types.
- **Command Groups.** This topic contains all the commands listed in functional groups. Each group consists of an overview of the commands in that group and a table that lists all the commands and queries for that group. You can click a command in the listing to display a detailed description of the command.
- **Status and Events.** This topic discusses the status and event reporting system for the GPIB interfaces. This system informs you of certain significant events that occur within the instrument. Topics that are discussed include registers, queues, event handling sequences, synchronization methods, and messages that the instrument may return, including error messages.
- **Miscellaneous.** This topic contains miscellaneous information, such as a list of reserved words, a table of the factory initialization (default) settings, and GPIB interface specifications that may be helpful when using GPIB commands to remotely control the instrument.

## Setting Up Remote Communications

Before setting up the instrument for remote communications using the electronic (physical) GPIB interface, you should familiarize yourself with the following GPIB requirements:

- A unique device address must be assigned to each device on the bus. No two devices can share the same device address.
- No more than 15 devices can be connected to any one line.
- One device should be connected for every 6 feet (2 meters) of cable used.
- No more than 65 feet (20 meters) of cable should be used to connect devices to a bus.
- At least two-thirds of the devices on the network should be powered on while using the network.
- Connect the devices on the network in a star or linear configuration. Do not use loop or parallel configurations.

### Connecting to the Instrument

Your instrument has a 24-pin GPIB connector on its rear (side) panel. This connector has a D-type shell and conforms to IEEE Std 488.1<sup>3</sup>/1987. Attach an IEEE Std 488.1<sup>3</sup>/1987 GPIB cable to this connector and to your controller as shown in the following figure.

If necessary, the GPIB connectors can be stacked as shown in the figure below.

### Setting the GPIB Address

To function correctly, your instrument must have a unique device address. The default settings for the GPIB configuration are:

- GPIB Address 1
- GPIB Mode GPIB Talk/Listen

To change either of the GPIB settings, do the following:

1. Select GPIB Configuration<sup>4</sup> from the Utilities menu.
2. Click the Configuration Talk/Listen button.
3. Change the GPIB Address to a unique address.
4. Click the Close button.

The instrument is now set up for bidirectional communication with your controller.



## Documentation

***DPO4000 Series Digital Phosphor Oscilloscopes User Manual.*** The user manual has information about installing and operating the instrument. It also provides concepts and theories about using the instrument..

***Getting Started with OpenChoice™ Solutions Manual.*** A book that explores some options for getting data from your instrument into any one of several available analysis tools.

***DPO4000 Series Digital Phosphor Oscilloscopes Specifications and Performance Verification.*** Instrument specifications and a performance verification procedure is available as a printable PDF file on the *DPO4000 Series Product Software* CD-ROM.

***TekVISA Programmer Manual.*** This manual is available as a printable PDF file on the *DPO4000 Series Product Software* CD-ROM. The manual describes TekVISA, the Tektronix implementation of the VISA Application Programming Interface (API). TekVISA is industry-compliant software for writing interoperable instrument drivers in a variety of Application Development Environments (ADEs).

***DPO4000 Series Digital Phosphor Oscilloscopes Service Manual.*** A printed service manual is available as an optional accessory. The service manual includes procedures to service the instrument to module levels. This manual is also available as a printable PDF file on the *DPO4000 Series Product Software* CD-ROM.

---



# Command Syntax

You can control the operations and functions of the instrument through the GPIB interface using commands and queries. The related topics listed below describe the syntax of these commands and queries. The topics also describe the conventions that the instrument uses to process them. See the *Command Groups* topic in the table of contents for a listing of the commands by command group, or use the index to locate a specific command.

## Backus-Naur Form Notation

This documentation describes the commands and queries using Backus-Naur Form (BNF) notation. Refer to the following table for the symbols that are used.

**Table 2-1: Symbols for Backus-Naur Form**

Symbol	Meaning
< >	Defined element
::=	Is defined as
	Exclusive OR
{ }	Group; one element is required
[ ]	Optional; can be omitted
. . .	Previous element(s) may be repeated
( )	Comment

## Command and Query Structure

Commands consist of set commands and query commands (usually called commands and queries). Commands modify instrument settings or tell the instrument to perform a specific action. Queries cause the instrument to return data and status information.

Most commands have both a set form and a query form. The query form of the command differs from the set form by its question mark on the end. For example, the set command `ACQuire:MODE` has a query form `ACQuire:MODE?`. Not all commands have both a set and a query form. Some commands have set only and some have query only.

## Messages

A command message is a command or query name followed by any information the instrument needs to execute the command or query. Command messages may contain five element types, defined in the following table.

**Table 2-2: Command Message Elements**

Symbol	Meaning
<Header>	This is the basic command name. If the header ends with a question mark, the command is a query. The header may begin with a colon (:) character. If the command is concatenated with other commands, the beginning colon is required. Never use the beginning colon with command headers beginning with a star (*).
<Mnemonic>	This is a header subfunction. Some command headers have only one mnemonic. If a command header has multiple mnemonics, a colon (:) character always separates them from each other.
<Argument>	This is a quantity, quality, restriction, or limit associated with the header. Some commands have no arguments while others have multiple arguments. A <space> separates arguments from the header. A <comma> separates arguments from each other.
<Comma>	A single comma is used between arguments of multiple–argument commands. Optionally, there may be white space characters before and after the comma.
<Space>	A white space character is used between a command header and the related argument. Optionally, a white space may consist of multiple white space characters.

**Commands**

Commands cause the instrument to perform a specific function or change one of the settings. Commands have the structure:

[ : ] <Header> [ <Space> <Argument> [ <Comma> <Argument> ] . . . ]

A command header consists of one or more mnemonics arranged in a hierarchical or tree structure. The first mnemonic is the base or root of the tree and each subsequent mnemonic is a level or branch off the previous one. Commands at a higher level in the tree may affect those at a lower level. The leading colon (:) always returns you to the base of the command tree.

**Queries** Queries cause the instrument to return status or setting information. Queries have the structure:

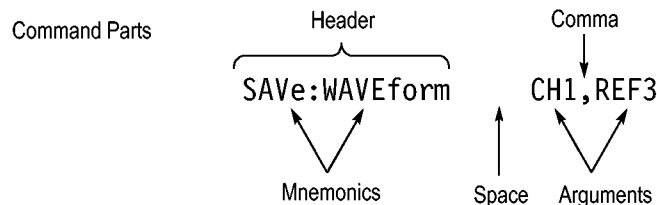
- [:]<Header>
- [:]<Header> [<Space><Argument> [<Coma><Argument>] ...]

You can specify a query command at any level within the command tree unless otherwise noted. These branch queries return information about all the mnemonics below the specified branch or level. For example, HISTogram:STATistics:STDdev? returns the standard deviation of the histogram, while HISTogram:STATistics? returns all the histogram statistics, and HISTogram? returns all the histogram parameters.

**Headers** You can control whether the instrument returns headers as part of the query response. Use the HEADER command to control this feature. If header is on, the query response returns command headers, then formats itself as a valid set command. When header is off, the response includes only the values. This may make it easier to parse and extract the information from the response. The table below shows the difference in responses.

**Table 2-3: Comparison of Header Off and Header On Responses**

Query	Header Off	Header On
TIME?	"14:30:00	:TIME"14:30:00
ACQuire:NUMAVg?	100	:ACQUIRE:NUMAVG 100



## Clearing the Instrument

You can clear the Output Queue and reset the instrument to accept a new command or query by using the selected Device Clear (DCL) GPIB function. Refer to your GPIB library documentation for further details about the selected Device Clear operation.

## Command Entry

The following rules apply when entering commands:

- You can enter commands in upper or lower case.
- You can precede any command with white space characters. White space characters include any combination of the ASCII control characters 00 through 09 and 0B through 20 hexadecimal (0 through 9 and 11 through 32 decimal).
- The instrument ignores commands consisting of any combination of white space characters and line feeds.

### Abbreviating

You can abbreviate many instrument commands. Each command in this documentation shows the abbreviations in capitals. For example, you can enter the command ACQuire:NUMAvg simply as ACQ:NUMA or acq:numa.

Abbreviation rules may change over time as new instrument models are introduced. Thus, for the most robust code, use the full spelling.

If you use the HEADer command to have command headers included as part of query responses, you can further control whether the returned headers are abbreviated or are full-length with the VERbose command.

### Concatenating

You can concatenate any combination of set commands and queries using a semicolon (;). The instrument executes concatenated commands in the order received.

When concatenating commands and queries, you must follow these rules:

1. Separate completely different headers by a semicolon and by the beginning colon on all commands except the first one. For example, the commands TRIGger:MODE NORMAl and ACQuire:NUMAVg 10, can be concatenated into the following single command:

```
TRIGger:MODE NORMAl;:ACQuire:NUMAVg 10
```

2. If concatenated commands have headers that differ by only the last mnemonic, you can abbreviate the second command and eliminate the beginning colon. For example, you can concatenate the commands ACQuire:MODE ENVelope and ACQuire:NUMAVg 10 into a single command:

```
ACQuire:MODE ENVelope; NUMAVg 10
```

The longer version works equally well:

```
ACQuire:MODE ENVelope;:ACQuire:NUMAVg 10
```

3. Never precede a star (\*) command with a colon:

```
ACQuire:MODE ENVelope;*OPC
```

Any commands that follow will be processed as if the star command was not there so the commands, `ACQuire:MODE ENvelope;*OPC;NUMAVg 10` will set the acquisition mode to envelope and set the number of acquisitions for averaging to 10.

4. When you concatenate queries, the responses to all the queries are concatenated into a single response message. For example, if the display imageview color is temperature and the display recordview color is spectral, the concatenated query `DISPlay:COLOR:PALETTE:IMAGEVIEW?;RECORDVIEW?` will return the following.

If the header is on:

```
:DISPlay:COLOR:PALETTE:IMAGEVIEW TEMPERATURE;
:DISPlay:COLOR:PALETTE:RECORDVIEW SPECTRAL
```

If the header is off:

```
TEMPERATURE;SPECTRAL
```

5. Set commands and queries may be concatenated in the same message. For example,

```
ACQuire:MODE SAMple;NUMAVg?;STATE?
```

is a valid message that sets the acquisition mode to sample. The message then queries the number of acquisitions for averaging and the acquisition state. Concatenated commands and queries are executed in the order received.

Here are some invalid concatenations:

```
DISPlay:STYLE:NORMAl;ACQuire:NUMAVg 10 (no colon before
ACQuire)
```

```
DISPlay:COLOR:CUSor1 1;;CUSor2 5 (extra colon before CUSor2;
use DISPlay:COLOR:CUSor1 1:CUSor2 5 instead)
```

```
DISPlay:STYLE:NORMAl;:*OPC (colon before a star (*) command)
```

```
DISPlay:COLOR:CUSor1 1;COLOR:CUSor2 5 (levels of the mnemonics
are different; either remove the second use of COLOR or place :DISPlay: in
front of COLOR:CUSor2 5)
```

## Terminating

This documentation uses `<EOM>` (End of message) to represent a message terminator.

**Table 2-4: End of Message Terminator**

Symbol	Meaning
<code>&lt;EOM&gt;</code>	Message terminator

The end-of-message terminator must be the END message (EOI asserted concurrently with the last data byte). The last data byte may be an ASCII linefeed (LF) character.

This instrument does not support ASCII LF only message termination. The instrument always terminates outgoing messages with LF and EOI.

## Constructed Mnemonics

Some header mnemonics specify one of a range of mnemonics. For example, a channel mnemonic can be CH1, CH2, CH3, or CH4. You use these mnemonics in the command just as you do any other mnemonic. For example, there is a CH1:POSition command, and there is also a CH2:POSition command. In the command descriptions, this list of choices is abbreviated as CH<x>.

### Cursor Position Mnemonics

When cursors are displayed, commands may specify which cursor of the pair to use.

**Table 2-5: Cursor Mnemonics**

Symbol	Meaning
CURSOR<x>	A cursor selector; <x> is either 1 or 2.
POSITION<x>	A cursor selector; <x> is either 1 or 2.
HPOS<x>	A cursor selector; <x> is either 1 or 2.

### Math Specifier Mnemonics

Commands can specify the mathematical waveform to use as a mnemonic in the header.

**Table 2-6: Math Specifier Mnemonics**

Symbol	Meaning
Math<x>	A math waveform specifier; <x> is 1 through 4.

### Measurement Specifier Mnemonics

Commands can specify which measurement to set or query as a mnemonic in the header. Up to eight automated measurements may be displayed.

**Table 2-7: Measurement Specifier Mnemonics**

Symbol	Meaning
MEAS<x>	A measurement specifier; <x> is 1 through 8.

### Channel Mnemonics

Commands specify the channel to use as a mnemonic in the header.



**Table 2-8: Channel Mnemonics**

Symbol	Meaning
CH<x>	A channel specifier; <x> is 1 through 4.

## Reference Waveform Mnemonics

Commands can specify the reference waveform to use as a mnemonic in the header.

**Table 2-9: Reference Waveform Mnemonics**

Symbol	Meaning
REF<x>	A reference waveform specifier; <x> is 1 thru 4.

## Argument Types

### Numeric

Many instrument commands require numeric arguments. The syntax shows the format that the instrument returns in response to a query. This is also the preferred format when sending the command to the instrument though any of the formats will be accepted. This documentation represents these arguments as follows:

**Table 2-10: Numeric Arguments**

Symbol	Meaning
<NR1>	Signed integer value
<NR2>	Floating point value without an exponent
<NR3>	Floating point value with an exponent
<bin>	Digital data in binary format
<hex>	Digital data in hexadecimal format

Most numeric arguments will be automatically forced to a valid setting, either by rounding or truncating, when an invalid number is input unless otherwise noted in the command description.

### Quoted String

Some commands accept or return data in the form of a quoted string, which is simply a group of ASCII characters enclosed by a single quote (') or double quote ("). The following is an example of a quoted string: "This is a quoted string". This documentation represents these arguments as follows:

**Table 2-11: Quoted String Argument**

Symbol	Meaning
<QString>	Quoted string of ASCII text

A quoted string can include any character defined in the 7-bit ASCII character set. Follow these rules when you use quoted strings:

1. Use the same type of quote character to open and close the string. For example: "this is a valid string".
2. You can mix quotation marks within a string as long as you follow the previous rule. For example, "this is an 'acceptable' string".
3. You can include a quote character within a string by repeating the quote. For example: "here is a "" mark".
4. Strings can have upper or lower case characters.
5. If you use a GPIB network, you cannot terminate a quoted string with the END message before the closing delimiter.
6. A carriage return or line feed embedded in a quoted string does not terminate the string, but is treated as just another character in the string.
7. The maximum length of a quoted string returned from a query is 1000 characters.

Here are some invalid strings:

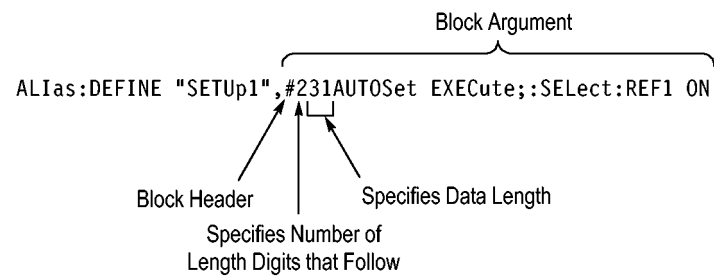
- "Invalid string argument' (quotes are not of the same type)
- "test<EOI>" (termination character is embedded in the string)

**Block** Several instrument commands use a block argument form (see the following table).

**Table 2-12: Block Argument**

Symbol	Meaning
<NZDig>	A nonzero digit character in the range of 1–9
<Dig>	A digit character, in the range of 0–9
<DChar>	A character with the hexadecimal equivalent of 00 through FF (0 through 255 decimal)
<Block>	A block of data bytes defined as: <Block> ::= {#<NZDig><Dig>[<Dig>...][<DChar>...]  #0[<DChar>...]<terminator>}

<NZDig> specifies the number of <Dig> elements that follow. Taken together, the <NZDig> and <Dig> elements form a decimal integer that specifies how many <DChar> elements follow.





# Command Groups

The DP04000 Series GPIB and RS-232 interfaces conform to Tektronix standard codes and formats except where noted. The GPIB interface also conforms to IEEE Std 488.2-1987 except where noted.

## Acquisition Command Group

Use the commands in the Acquisition Command Group to set up the modes and functions that control how the instrument acquires the signals you input to the channels and processes them into waveforms.

Using these commands for acquiring waveforms, you can do the following:

- Start and stop acquisitions.
- Control whether each waveform is simply acquired, averaged, or enveloped over successive acquisitions of that waveform.
- Set the controls or conditions that start and stop acquisitions.
- Determine the action the system takes upon completing an acquisition, such as saving all waveforms and taking a measurement when the acquisition is stopped.
- Control acquisition of acquired channel waveforms.
- Set acquisition parameters.

**Table 2-13: Acquisition Commands**

Command	Description
<a href="#">ACQuire?</a>	Sets or returns acquisition mode
<a href="#">ACQuire:MAXSamplerate?</a>	Returns the maximum real-time sample rate
<a href="#">ACQuire:MODE</a>	Returns the number of acquisitions that have occurred
<a href="#">ACQuire:NUMACq?</a>	Sets or returns number of acquisitions for an averaged waveform
<a href="#">ACQuire:NUMAVg</a>	Sets or returns number of acquisitions for envelope waveform
<a href="#">ACQuire:NUMEnv</a>	Sets or returns the number of samples that make up a WfmDB for single sequence mode and Mask Pass/Fail Completion Test
<a href="#">ACQuire:STATE</a>	Sets or returns whether the acquisition is continuous or single sequence
<a href="#">ACQuire:STOPAfter</a>	Returns acquisition parameters

**Table 2-13: Acquisition Commands, (cont.)**

Command	Description
<a href="#">FASTAcq?</a>	Enables, disables, or returns state of Fast Acquisition mode
<a href="#">FASTAcq:STATE</a>	Returns the Fast Acquisition state

## Alias Command Group

Alias commands allow you to define new commands as a sequence of standard commands. You may find this useful when repeatedly using the same commands to perform certain tasks like setting up measurements.

Aliases are similar to macros but do not include the capability to substitute parameters into alias bodies. The alias mechanism obeys the following rules:

- The alias name must consist of a valid IEEE 488.2 message unit, which may not appear in a message preceded by a colon, comma, or a command or query program header.
- The alias name may not appear in a message followed by program date, a colon, comma, or question mark.
- An alias name must be distinct from any keyword or keyword short form.
- An alias name cannot be redefined without first being deleted using one of the alias deletion functions.
- Alias names do not appear in response messages.
- The Alias commands are defined in Tektronix Standard Codes and Formats. Deviations between that standard and what is specified here will be considered DPO4000 series errors unless specifically noted in the command description in this document.

**Table 2-14: Alias Commands**

Command	Description
<a href="#">ALias</a>	Sets or returns the alias state
<a href="#">ALias:CATalog?</a>	Returns a list of the currently defined alias labels
<a href="#">ALias:DEFine</a>	Assigns a sequence of program messages to an alias label
<a href="#">ALias:DELEte</a>	Removes a specified alias
<a href="#">ALias:DELEte:ALL</a>	Deletes all existing aliases

Table 2-14: Alias Commands, (cont.)

Command	Description
<a href="#">ALias:DELEte[:NAME]</a>	Removes a specified alias
<a href="#">ALias[:STATE]</a>	Sets or returns the alias state

## Calibration and Diagnostic Command Group

The Calibration and Diagnostic commands provide information about the current state of instrument calibration and allow you to initiate internal signal path calibration (SPC) or execute diagnostic tests. Commands that are specific to factory calibration are not described in this manual; however, they are described in the service manual (located on your DPO4000 Documentation CD-ROM in PDF format). You can also order a printed copy.

Table 2-15: Calibration and Diagnostic Commands

Command	Description
<a href="#">*CAL?</a>	Instructs the instrument to perform self-calibration
<a href="#">CAL?</a>	Returns the internal and factory calibration status and the calibration due date
<a href="#">CALibrate:FACTory</a>	Starts and stops the factory calibration process
<a href="#">CALibrate:FACTory:NOTIfy:DUE?</a>	Queries if calibration is due
<a href="#">CALibrate:FACTory:NOTIfy:HOuRs</a>	Sets or returns hours when calibration is due
<a href="#">CALibrate:FACTory:NOTIfy:YEARS</a>	Sets or returns the number of years when calibration is due
<a href="#">CALibrate:FACTory:STATus?</a>	Returns the factory calibration status value saved in nonvolatile memory
<a href="#">CALibrate:FACTory:STEPSTATus?</a>	Returns information to synchronize programmed factory calibration steps
<a href="#">CALibrate:FACTory:STEPSTIMulus?</a>	Returns information about an input signal required for calibration
<a href="#">CALibrate:INTERNaL</a>	Starts the internal signal path calibration
<a href="#">CALibrate:INTERNaL:STARt</a>	Starts the internal signal path calibration
<a href="#">CALibrate:INTERNaL:STATus?</a>	Returns the current status of the internal signal path calibration.
<a href="#">CALibrate:RESults?</a>	Returns the status of all calibration subsystems without performing an SPC operation
<a href="#">CALibrate:RESults:SPC?</a>	Returns the results of the last SPC operation
<a href="#">CALibrate:RESults:FACTory?</a>	Returns the status of internal and factory calibration

**Table 2-15: Calibration and Diagnostic Commands, (cont.)**

Command	Description
<a href="#">CALibrate:RESults:FACtory:Frequency?</a>	Returns the status of internal and factory frequency calibration
<a href="#">CALibrate:RESults:FACtory:Trigger?</a>	Returns the status of internal and factory trigger calibration
<a href="#">CALibrate:RESults:FACtory:Voltage?</a>	Returns the status of internal and factory voltage calibration
<a href="#">CALibrate:TEMPerature?</a>	Returns the temperature during the last signal path compensation operation
<a href="#">DIAg:LOOP:OPTion</a>	Sets the diagnostic loop option
<a href="#">DIAg:LOOP:OPTion:NTIMes</a>	Sets the diagnostic loop option to run N times
<a href="#">DIAg:LOOP:STOP</a>	Stops diagnostic at the end of the current loop
<a href="#">DIAg:MODe</a>	Select the test suite of diagnostics that is to be run
<a href="#">DIAg:RESUlt:FLAg?</a>	Returns the pass/fail status from the last diagnostic test sequence execution
<a href="#">DIAg:RESUlt:LOG?</a>	Returns the internal results log from the last diagnostic test sequence execution
<a href="#">DIAg:SElect:ALL</a>	Selects all available diagnostics
<a href="#">DIAg:SElect:&lt;function&gt;</a>	Selects one of the available diagnostic areas
<a href="#">DIAg:STATE</a>	Sets the instrument operating state

## Cursor Command Group

Use the commands in the Cursor Command Group to control the cursor display and readout. You can use these commands to control the setups for cursor 1 and cursor 2, such as waveform source, cursor position, and cursor color.



You can also use the commands to select one of the following cursor functions:

- **Off** Shuts off the display of all cursors.
- **Vertical Bars.** Displays vertical bar cursors, which provide traditional horizontal unit readouts for Cursor 1 (bar1), Cursor 2 (bar2), the delta between them, and 1/delta (results in frequency when the horizontal unit is time).
- **Horizontal Bars.** Displays horizontal bar cursors, which provide traditional vertical unit readouts for Cursor 1 (bar1), Cursor 2 (bar2), and the delta between them.
- **Waveform Cursors.** Consists of two cursors you can independently assign to a waveform. These cursors provide the same readouts that the vertical and horizontal bar cursors provide. Waveform cursors enable you to conveniently measure waveform amplitude and time. In XY or XYZ format, waveform cursors indicate the amplitude position of an XY pair (Ch1 vs Ch2 voltage, where Ch1 is the X axis and Ch2 is the Y axis) relative to the trigger.
- **Screen Cursors.** Consists of two pairs of independent horizontal and vertical cursors. You can use these cursors to indicate an arbitrary position within the waveform display area. Screen cursors, depending on the style selected, consist of the intersection of a vertical and horizontal line, an X, or a vertical line with an X. These cursors have no association with any waveform, other than they inherit the color of the waveform they are assigned too.

**Table 2-16: Cursor Commands**

Command	Description
<a href="#">CURSor?</a>	Returns all cursor settings
<a href="#">CURSor:FUNCTioN</a>	Sets or returns the cursor type
<a href="#">CURSor:HBArs?</a>	Returns hbar cursor settings
<a href="#">CURSor:HBArs:DELTA?</a>	Returns hbars cursors vertical difference
<a href="#">CURSor:HBArs:POSITION&lt;x&gt;</a>	Sets or returns the hbar cursor<x> vertical position
<a href="#">CURSor:HBArs:UNIts</a>	Returns hbar cursor units
<a href="#">CURSor:HBArs:USE</a>	Sets the horizontal bar cursor measurement scale
<a href="#">CURSor:MODe</a>	Sets or returns whether cursors move in unison or separately
<a href="#">CURSor:VBArS?</a>	Sets or returns the position of vertical bar cursors
<a href="#">CURSor:VBArS:DELTA?</a>	Returns the difference between vbar cursors
<a href="#">CURSor:VBArS:HPOS&lt;x&gt;?</a>	Returns the horizontal value of the specified vertical bar ticks
<a href="#">CURSor:VBArS:POSITION&lt;x&gt;</a>	Sets or returns the vbar cursor<x> horizontal position
<a href="#">CURSor:VBArS:UNIts</a>	Sets or returns the units for vbar cursors

**Table 2-16: Cursor Commands, (cont.)**

Command	Description
<a href="#">CURSor:VBArS:USE</a>	Sets the vertical bar cursor measurement scale
<a href="#">CURSor:VBArS:VDELtA?</a>	Returns the vertical difference between the two vertical bar cursor ticks

## Display Command Group

Use the commands in the Display command group to change the graticule style, the displayed intensities, and to set the characteristics of the waveform display.

You can set the display of date and time; cursor, histogram, mask, and measurement readouts; measurement annotations, and the mode in which waveforms are displayed.

There are six color palettes from which you can select:

- **Normal** displays hues and lightness levels for best overall viewing.
- **Temp** displays areas of the waveform with the highest sample density in warmer colors (red shades) while the areas of lowest sample density appear in cooler colors (blue shades).
- **Spectral** displays areas of the waveform with the highest sample density in blue shades while the areas of lowest sample density appear in red shades.
- **Green** displays waveforms in shades of green. Areas of the waveform with the highest sample density appear in lighter green shades while the areas of lowest sample density appear in darker green shades.
- **Gray** displays waveforms in shades of gray. Areas of the waveform with the highest sample density appear in lighter gray shades while the areas of lowest sample density appear in darker gray shades.
- **User** allows you to create a customized color palette.

Use the commands to set the style that best displays your waveforms and graticule display properties. Note that the mode you choose globally affects all displayed waveforms.

**Table 2-17: Display Commands**

Command	Description
<a href="#">DISplay?</a>	Returns current display settings
<a href="#">DISplay:CLOCK</a>	Sets or returns the display of the date/time stamp
<a href="#">DISplay:FORMat</a>	Sets or returns the display format

Table 2-17: Display Commands, (cont.)

Command	Description
DISplay:GRAticule	Sets or returns the type of graticule that is displayed
DISplay:INTENSITy?	Returns all display intensity settings
DISplay:INTENSITy:BACKLight	Sets or returns the backlight intensity for the display
DISplay:INTENSITy:GRAticule	Sets or returns the graticule intensity for the display
DISplay:INTENSITy:WAVEform	Sets or returns the intensity of the waveforms
DISplay:PERSiStence	Sets or returns display persistence setting
DISplay:PICTure:AUTOContrast	Sets or returns the video picture mode autocontrast setting
DISplay:PICTure:BRiGhtness	Sets or returns the video picture mode brightness setting
DISplay:PICTure:CONTRAst	Sets or returns the video picture mode contrast setting
DISplay:PICTure:STATE	Sets or returns the video picture mode setting
DISplay:STYle:DOTsonly	Sets a dots-only display
DISplay:VECTorscope:STATE	Sets or returns the video vectorscope state setting
DISplay:VECTorscope:STsetting	
DISplay:VECTorscope:TYPe	Sets or returns the video vectorscope display type setting
DISplay:VECTorscope:TYsetting	
DISplay:XY:MODe	Sets or returns the display mode
DISplay:XY:YCHannel	Specifies the Y channel to be displayed against a channel waveform
DISplay:XY:YREF	Specifies the Y channel to be displayed against a reference waveform
MESSage:BOX	Defines the size and position of the message window
MESSage:CLEAR	Removes the message text from the message window

**Table 2-17: Display Commands, (cont.)**

Command	Description
<a href="#">MESSage:SHOW</a>	Clears the contents of the message window and displays the new message in the window
<a href="#">MESSage:STATE</a>	Controls the display of the message window

## Ethernet Command Group

Use the commands in the Ethernet Command group to set up the Ethernet remote interface.

**Table 2-18: Ethernet Commands**

Command	Description
<a href="#">ETHERnet:DHCPbootp</a>	Sets or returns the network initialization search for a DHCP/BOOTP server
<a href="#">ETHERnet:DNS:IPADdress</a>	Sets or returns the network Domain Name Server (DNS) IP address
<a href="#">ETHERnet:DOMAINname</a>	Sets or returns the network domain name
<a href="#">ETHERnet:ENET:ADdress?</a>	Returns the Ethernet address value assigned to the oscilloscope
<a href="#">ETHERnet:GATEWay:IPADdress</a>	Sets or returns the remote interface gateway IP address
<a href="#">ETHERnet:HTTTPort</a>	Sets or returns the remote interface HTTP port value
<a href="#">ETHERnet:IPADdress</a>	Sets or returns the IP address assigned to the oscilloscope
<a href="#">ETHERnet:NAME</a>	Sets or returns the network name assigned to the oscilloscope
<a href="#">ETHERnet:PASSWord</a>	This command sets or returns the Ethernet access password
<a href="#">ETHERnet:SUBNETMask</a>	Sets or returns the remote interface subnet mask value

## File System Command Group

Use the commands in the File System Command Group to help you use the built-in hard disk drive and floppy disk drive. You can use the commands to do the following:

- List the contents of the default directory
- Create and delete directories
- Create, copy, read, rename, or delete a file

When using these commands, keep the following points in mind:

- File arguments are always enclosed within double quotes:  
"C:\MYDIR\TEK00001.SET"
- File names follow the MSDOS format: [DRIVE:][\PATH\]filename
- Path separators may be either forward slashes (/) or back slashes (\)

---

**NOTE.** Using back slash as a path separator may produce some unexpected results, depending on how your GPIB controller application treats escaped characters. Many applications recognize the sequence of back slash followed by an alphabetic character as an escaped character, and, as such, interpret that alphabetic character as a control character. For example, the sequence "\n" may be interpreted as a newline character; "\t" may be interpreted as a tab character. To ensure that this interpretation does not occur, you can use double back slashes. For example, "C:\\testfile.txt".

---

- Some FILESystem commands may fail because a file has read-only attributes. You will not be able to delete or replace such files until this attribute is removed..

**Table 2-19: File System Commands**

Command	Description
FILESystem?	Returns the file system state
FILESystem:COpy	Copies one or more files to a new file
FILESystem:CWD	Sets or returns the current working directory for FILESystem GPIB commands.
FILESystem:DELEte	Deletes a named file or directory
FILESystem:DIR?	Returns a list of directory contents
FILESystem:FREEspace?	Returns the number of bytes of free space on the current drive
FILESystem:MKDir	Makes a new directory
FILESystem:REName	Assigns a new name to an existing file
FILESystem:RMDir	Deletes the named directory

## Hard Copy Command Group

Hard copy commands enable you to make hard copies.

**Table 2-20: Hard Copy Commands**

Command	Description
<a href="#">HARDCopy</a>	Sends a screen copy to the selected port or returns the selected port and file path
<a href="#">HARDCopy:ACTIVEprinter</a>	Sets or returns the currently active printer
<a href="#">HARDCopy:INKSaver</a>	Changes hard copy output to print color traces and graticule on a white background
<a href="#">HARDCopy:LAYout</a>	Sets or returns the page orientation for hard copy
<a href="#">HARDCopy:PREVIEW</a>	Previews the current screen contents with the InkSaver palette applied
<a href="#">HARDCopy:PRINTer:ADD</a>	Adds a network printer to the list of available printers
<a href="#">HARDCopy:PRINTer:DELeTe</a>	Removes a network printer from the list of available printers
<a href="#">HARDCopy:PRINTer:LIST?</a>	Returns the list of currently attached printers
<a href="#">HARDCopy:PRINTer:REName</a>	Renames a network printer on the list of available printers

## Horizontal Command Group

Horizontal commands control the time bases of the instrument. You can set the time per division (or time per point) of the main time base. You can use the Horizontal commands to do the following:

- Set the scale, horizontal position and reference, and units of the time base
- Get the screen resolution, time of first point and time of last point, or get all the horizontal settings
- Enable or disable the display of the time base

You may substitute SECdiv for SCALE in the horizontal commands. This provides program compatibility with earlier models of Tektronix instruments.

**Table 2-21: Horizontal Commands**

Command	Description
<a href="#">HORizontal?</a>	Returns all learnable settings for the horizontal commands
<a href="#">HORizontal:ACQLENGTH?</a>	Returns the record length
<a href="#">HORizontal:MAIn?</a>	Returns the time per division of the time base
<a href="#">HORizontal[:MAIn]:DELay:MODe</a>	Sets or returns the main time base trigger delay mode
<a href="#">HORizontal[:MAIn]:DELay:STATe</a>	Sets or returns the time base trigger delay mode

Table 2-21: Horizontal Commands, (cont.)

Command	Description
<a href="#">HORizontal[:MAIn]:DELay:TIME</a>	Sets or returns the main time base trigger delay time
<a href="#">HORizontal[:MAIn]:POSition?</a>	Sets or returns the waveform horizontal position on the display
<a href="#">HORizontal:MAIn:SAMPLERate?</a>	Sets the horizontal sample rate to the desired number of samples per second Or returns the current horizontal sample rate
<a href="#">HORizontal[:MAIn]:SCALE</a>	Sets time per division for the main time base Or returns the main time base horizontal scale
<a href="#">HORizontal:MAIn:SECdiv</a>	Sets time per division for the main time base Or returns the main time base horizontal scale
<a href="#">HORizontal:MAIn:UNIts?</a>	Returns the units for the horizontal main time base
<a href="#">HORizontal:MAIn:UNIts:STRing?</a>	Sets or returns the units string for the horizontal main time base
<a href="#">HORizontal:PREViewstate?</a>	Returns whether or not the acquisition system is in the preview state
<a href="#">HORizontal:RECOrdlength</a>	Sets the horizontal record length to the number of data points in each frame Or returns the current horizontal record length
<a href="#">HORizontal:RESOLution</a>	Sets the horizontal record length to the number of data points in each frame and simultaneously adjusts the sample rate to maintain a constant time/division Or returns the current horizontal record length
<a href="#">HORizontal:ROLL?</a>	Returns the horizontal roll mode status

## Mark Command Group

The Mark commands provide...

### Mark Commands

Command	Description
<a href="#">MARK</a>	Move to the next or previous mark on the waveform.
<a href="#">MARK:CREATE</a>	Creates a mark on a particular waveform or all waveforms in a column.

**Mark Commands, (cont.)**

Command	Description
<a href="#">MARK:DELEte</a>	Deletes a mark on a particular waveform, all waveforms in a column, or all marks.
<a href="#">MARK:FREE?</a>	Returns how many marks are free to be used
<a href="#">MARK:SElected:END?</a>	Returns the end of the selected mark, in terms of 0 to 100% of the waveform.
<a href="#">MARK:SElected:FOCUS?</a>	Returns the focus of the selected mark, in terms of 0 to 100% of the waveform.
<a href="#">MARK:SElected:MARKSINCOLumn?</a>	Returns how many marks are in the current zoom pixel column.
<a href="#">MARK:SElected:OWNer?</a>	Returns the owner of the selected mark.
<a href="#">MARK:SElected:SOURCE?</a>	Returns the source waveform of the selected mark.
<a href="#">MARK:SElected:START?</a>	Returns the start of the selected mark, in terms of 0 to 100% of the waveform.
<a href="#">MARK:SElected:STATe?</a>	Returns the on or off state of the selected mark.
<a href="#">MARK:SElected:ZOOm:POSition?</a>	Returns the position of the selected mark, in terms of 0 to 100% of the upper window.
<a href="#">MARK:TOTal?</a>	Returns how many marks are used.

## Math Command Group

Use the commands in the Math Command Group to create and define math waveforms. You can define and display up to four math waveforms simultaneously on four channel models and up to two on two channel models. Use the available math functions to define your math waveform.

The math waveform you create depends on sources listed in the math expression. If you change these sources, the math waveforms you previously defined will be affected.

Math expressions can be simple, containing no mathematical computation, such as CH1, which specifies that a waveform shows the signal source of channel 1. Math expressions can also be complex, consisting of 100 plus characters and comprising many sources, functions, and operands.

The acquisition of a live waveform can stop for several reasons: You can turn off the channel, stop the waveform (via Run/Stop from the Horiz/Acq menu), or stop the trigger (via Run/Stop from the Trig menu). When you turn off the channel, math continues and data is acquired but is not displayed. When you stop either the waveform or the trigger, the math calculation stops, and the last math calculation performed is displayed.



When a live waveform update or reference waveform is altered, math waveforms containing those waveforms as sources are also updated to reflect the changes. Also, sources must exist but do not need to be displayed to be used in and to update math waveforms.

**Table 2-22: Math Commands**

Command	Description
MATH[1]:DEFine	Sets or returns the math<x> waveform definition for the specified waveform
MATH[1]:HORizontal:POSition	Sets or returns the math horizontal display position math waveforms
MATH[1]:HORizontal:SCAle	Sets or returns the math horizontal display scale for math waveforms
MATH[1]:HORizontal:UNITs	Sets or returns the math waveform horizontal measurement unit value
MATH[1]:SPECTral:MAG	Sets or returns the units of the SpectralMag function in the specified math string
MATH[1]:SPECTral:WINDow	Sets or returns the window function used to multiply the input data to the spectral analyzer
MATH[1]:TYPE	Sets or returns the math waveform mode type
MATH[1]:VERTical:POSition	Sets or returns the math waveform vertical position
MATH[1]:VERTical:SCAle	Sets or returns the math waveform vertical scale (per div)
MATH[1]:VERTical:UNITs	Sets or returns the units used for the math waveform; command overrides the default unit string
MATHVAR?	Returns all numerical values used within math expressions
MATHVAR:VAR<x>	Sets or returns numerical values you can use within math expressions

## Measurement Command Group

Use the commands in the Measurement Command Group to control the automated measurement system.

Up to eight automated measurements can be displayed on the screen. In the commands, these measurement readouts are named MEAS<x>, where <x> can be 1 through 8.

In addition to the eight displayed measurements, the measurement commands let you specify an additional measurement, IMMEd. The immediate measurement

has no front-panel equivalent. Immediate measurements are never displayed. Because they are computed only when needed, immediate measurements slow the waveform update rate less than displayed measurements.

Whether you use displayed or immediate measurements, use the VALue query to obtain measurement results.

Measurement commands can set and query measurement parameters. You can assign some parameters, such as waveform sources, differently for each measurement. Other parameters, such as reference levels, have only one value, which applies to all measurements.

**Table 2-23: Measurement Commands**

Command	Description
MEASUrement?	Returns all measurement parameters
MEASUrement:CLEARSnapshot	Removes the measurement snapshot display
MEASUrement:GATing	Sets or returns the measurement gating
MEASUrement:IMMed?	Returns all immediate measurement setup parameters
MEASUrement:IMMed:DElay?	Returns information about the immediate delay measurement
MEASUrement:IMMed:DElay:DIREction	Sets or returns the search direction to use for immediate delay measurements
MEASUrement:IMMed:DElay:EDGE<x>	Sets or returns the slope of the edge used for immediate delay “from” and “to” waveform measurements
MEASUrement:IMMed:SOURCE	Sets or returns the “from” source for all single channel immediate measurements
MEASUrement:IMMed:SOURCE2	Sets or returns the source to measure “to” for phase or delay immediate measurements
MEASUrement:IMMed:TYPE	Sets or returns the type of the immediate measurement
MEASUrement:IMMed:UNIts?	Returns the units of the immediate measurement
MEASUrement:IMMed:VALue?	Returns the value of the immediate measurement
MEASUrement:INDICators?	Returns all measurement indicator parameters
MEASUrement:INDICators:HORIZ<x>?	Returns the position of the specified horizontal measurement indicator
MEASUrement:INDICators:NUMHORIZ?	Returns the number of horizontal measurement indicators currently being displayed
MEASUrement:INDICators:NUMVERT?	Returns the number of vertical measurement indicators currently being displayed

Table 2-23: Measurement Commands, (cont.)

Command	Description
MEASUrement:INDICators:STATE	Set or returns the state of visible measurement indicators
MEASUrement:INDICators:VERT<x>?	Returns the value of the specified vertical measurement indicator
MEASUrement:MEAS<x>?	This query-only command returns all measurement parameters
MEASUrement:MEAS<x>:COUNT?	Returns the number of values accumulated since the last statistical reset
MEASUrement:MEAS<x>:DElay?	Returns the delay measurement parameters for specified measurement
MEASUrement:MEAS<x>:DElay:DIREction	Sets or returns the search direction to use for delay measurements
MEASUrement:MEAS<x>:DElay:EDGE<x>	Sets or returns the slope of the edge to use for delay “from” and “to” waveform measurements
MEASUrement:MEAS<x>:MAXimum?	Returns the maximum value found since the last statistical reset
MEASUrement:MEAS<x>:MEAN?	Returns the mean value accumulated since the last statistical reset
MEASUrement:MEAS<x>:MINimum?	Returns the minimum value found since the last statistical reset
MEASUrement:MEAS<x>:SOURCE[1]	Sets or returns the channel from which measurements are taken
MEASUrement:MEAS<x>:SOURCE2	Sets or returns the channel to which measurements are sent
MEASUrement:MEAS<x>:STATE	Sets or returns whether the specified measurement slot is computed and displayed
MEASUrement:MEAS<x>:STDdev?	Returns the standard deviation of values accumulated since the last statistical reset
MEASUrement:MEAS<x>:TYPE	Sets or returns the measurement<x> type
MEASUrement:MEAS<x>:UNIts?	Returns measurement<x> units
MEASUrement:MEAS<x>:VALue?	Returns the value of measurement<x>
MEASUrement:METHod	Sets or returns the method used for calculating reference levels
MEASUrement:REFLevel?	Returns the current reference level parameters
MEASUrement:REFLevel:ABSolute:HIGH	Sets or returns the top reference level for rise time
MEASUrement:REFLevel:ABSolute:LOW	Sets or returns the low reference level for rise time

**Table 2-23: Measurement Commands, (cont.)**

Command	Description
<a href="#">MEASUrement:REFLevel:ABSolute:MID</a>	Sets or returns the mid reference level for measurements
<a href="#">MEASUrement:REFLevel:ABSolute:MID2</a>	Sets or returns the mid reference level for delay "to" measurements
<a href="#">MEASUrement:REFLevel:METHod</a>	Sets or returns the method for assigning high and low reference levels
<a href="#">MEASUrement:REFLevel:PERCent:HIGH</a>	Sets or returns the top reference percent level for rise time
<a href="#">MEASUrement:REFLevel:PERCent:LOW</a>	Sets or returns the low reference percent level for rise time
<a href="#">MEASUrement:REFLevel:PERCent:MID</a>	Sets or returns the mid reference percent level for waveform measurements
<a href="#">MEASUrement:REFLevel:PERCent:MID2</a>	Sets or returns the mid reference percent level for second waveform measurements
<a href="#">MEASUrement:SNAPShot</a>	Displays the measurement snapshot list
<a href="#">MEASUrement:STATistics:MODE</a>	Turns management statistics on or off and sets or returns which pair of statistical accumulations is displayed
<a href="#">MEASUrement:STATistics:WEIghing</a>	Sets or returns the 'time constant' for mean and standard deviation statistical accumulations

## Miscellaneous Command Group

Miscellaneous commands do not fit into other categories.

Several commands and queries are common to all 488.2-1987 devices on the GPIB bus. The 488.2-1987 standard defines these commands. The common commands begin with an asterisk (\*) character.

**Table 2-24: Miscellaneous Commands**

Command	Description
<a href="#">AUTOSet</a>	Sets the vertical, horizontal and trigger controls to provide a stable display of the selected waveform
<a href="#">AUXout?</a>	Returns the auxiliary out setup
<a href="#">AUXout:EDGE</a>	Sets or returns the polarity of the auxiliary output
<a href="#">AUXout:SOUrce</a>	Sets or returns the trigger source at the BNC connection

Table 2-24: Miscellaneous Commands, (cont.)

Command	Description
BELI	In previous models, this command sounded the audio indicator. Provided for backward compatibility.
CLEARMenu	Clears the current menu from the display
DATE	Sets or returns the date that the instrument can display
*DDT	Sets or returns the commands that will be executed by the group execute trigger
FPANEL:PRESS	Duplicates the action of pressing a specified front-panel button
FPANEL:TURN	Duplicates the action of turning a specified front-panel control knob
GPIBUsb:ADdResS?	Returns the GPIB/USB address.
GPIBUsb:HWVersion	Sets or returns the GPIB/USB hardware version.
GPIBUsb:ID?	Sets or returns the GPIB/USB ID.
GPIBUsb:SETADdResS	
GPIBUsb:SETID	
GPIBUsb:STATUS	
HDR	Sets or returns the Response Header Enable State
HEADer	Sets or returns the Response Header Enable State
ID?	Returns identifying information about the instrument and its firmware
*IDN?	Returns the instrument identification code
LANGuage	Sets or returns the user interface display language
LANGuage:INCRement	Changes the oscilloscope display language to the next language in the sequence
LOCK	Sets or returns the front panel lock state
*LRN?	Returns a listing of instrument settings
NEWpass	Changes the password for user protected data
REM	Specifies a comment, which is ignored by the instrument
SET?	Returns a listing of instrument settings
STArtuptime?	Returns the time that the oscilloscope was last powered on
TEKSecure	Initializes both waveform and setup memories

**Table 2-24: Miscellaneous Commands, (cont.)**

Command	Description
TIME	Sets or returns the time displayed by the instrument
TOTaluptime?	Returns the total number of hours that the oscilloscope has been turned on since the nonvolatile memory was last programmed
*TRG	Performs the group execute trigger (GET)
*TST?	Tests the GPIB interface and returns status
UNLock	Unlocks front panel
VERBose	Sets or returns the verbose state

## Save and Recall Command Group

Use the commands in the Save and Recall Command Group to store and retrieve internal waveforms and settings. When you save a setup, you save all the settings of the instrument. When you recall a setup, the instrument restores itself to the state that it was in when you originally saved that setting.

**Table 2-25: Save and Recall Commands**

Command	Description
FACTory	Resets the instrument to factory default settings
*RCL	Restores the state of the instrument from a copy of the setting stored in memory
RECAI:SETUp	Recalls saved instrument settings
RECAI:WAVEform	Recalls a stored waveform to a reference location
*SAV	Stores the state of the instrument to a specified memory location
SAVe:IMAGe	Saves a capture of the screen image into the specified file
SAVe:IMAGe:FILEFormat	Sets or returns the format for images
SAVe:SETUp	Saves the current front-panel setup to a specified memory location or file

Table 2-25: Save and Recall Commands, (cont.)

Command	Description
SAVe:WAVEform	Saves a waveform to one of four reference memory locations or a mass storage file
SAVe:WAVEform:FILEFormat	Sets or returns the format for saved waveforms

## Search Command Group

The Search commands provide...

### Search Commands

Command	Description
SEARCH:SEARCH<x>:COPy	Copies the search criteria to the trigger, or the trigger criteria to the search.
SEARCH:SEARCH<x>:STATE	Sets the search state to on or off
SEARCH:SEARCH<x>:TOTAL?	Returns the total amount of matches for this search
SEARCH:SEARCH<x>:TRIGger:A:BUS	Sets or queries the serial search type
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:CONDition	Sets or queries the search condition for CAN search
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:DATA:DIRection	Sets or queries the CAN search condition to be valid on a READ, WRITE or either
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:DATA:QUALifier	Sets or queries the CAN data qualifier
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:DATA:SIZE	Sets or queries the length of the data string in bytes to be used for CAN search
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:DATA:VALue	Sets or queries the binary data string to be used for CAN search
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:FRAMEtype	Sets or queries the CAN Frame Type to be used
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN{:IDentifier}:ADDRess}:MODE	Sets or queries the CAN addressing mode to standard or extended format
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN{:IDentifier}:ADDRess}:VALue	sets or queries the binary address string to be used for CAN search
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:ADDRess:MODE	Sets or queries the I2C address mode to 7 or 10-Bit
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:ADDRess:TYPE	Sets or queries the I2C address type to I2C special addresses
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:ADDRess:VALue	Sets or queries the binary address string to be used for I2C search

## Search Commands, (cont.)

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:CONDition	Sets or queries the search condition for I2C search
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:DATA:DIRection	Sets or queries the I2C search condition to be valid on a READ, WRITE or either
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:DATA:SIZE	Sets or queries the length of the data string in bytes to be used for I2C search
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:DATA:VALue	Sets or queries the binary data string to be used for I2C search
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:SPI:CONDition	Sets or queries the search condition for SPI search
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:SPI:DATA{:MISO :IN}:VALue	Sets or queries the binary data string to be used for SPI search if
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:SPI:DATA{:MOSI :OUT}:VALue	Sets or queries the binary data string to be used for SPI search
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:SPI:DATA:SIZE	Sets or queries the length of the data string in bytes to be used for SPI search
SEARCH:SEARCH<x>:TRIGger:A:BUS:SOUrce	Sets or queries the bus for a serial search
SEARCH:SEARCH<x>:TRIGger:A:EDGE:SLOpe	Sets or returns the slope for an edge search
SEARCH:SEARCH<x>:TRIGger:A:EDGE:SOUrce	Sets or returns the source waveform for an edge search
SEARCH:SEARCH<x>:TRIGger:A:LEVel	Sets or queries the level for an edge search
SEARCH:SEARCH<x>:TRIGger:A:LEVel:CH<x>	Sets or queries the level for edge search for the selected channel
SEARCH:SEARCH<x>:TRIGger:A:LEVel:MATH	Sets or queries the math waveform level for edge search
SEARCH:SEARCH<x>:TRIGger:A:LEVel:REF<x>	Sets or queries the reference waveform level for edge search
SEARCH:SEARCH<x>:TRIGger:A:LOGic:FUNCTion	Sets or queries the logic operator for the logic search
SEARCH:SEARCH<x>:TRIGger:A:LOGic:INPut:CH<x>	Sets or queries the Boolean logic criteria for the logic search
SEARCH:SEARCH<x>:TRIGger:A:LOGic:INPut:CLOCK:EDGE	Sets or queries whether the clock edge is rise or fall for a logic search
SEARCH:SEARCH<x>:TRIGger:A:LOGic:INPut:CLOCK:SOUrce	Sets or queries the clock source definition for logic search
SEARCH:SEARCH<x>:TRIGger:A:LOGic:INPut:MATH	Sets or queries the Boolean logic criteria for the logic search
SEARCH:SEARCH<x>:TRIGger:A:LOGic:INPut:REF<x>	Sets or queries the Boolean logic criteria for the logic search



## Search Commands, (cont.)

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:LOGic:PAter:n:INPut:CH<x>	Sets or queries the Boolean logic criteria for the logic search
SEARCH:SEARCH<x>:TRIGger:A:LOGic:PAter:n:INPut:MATH	Sets or returns the Boolean logic criteria for the logic search
SEARCH:SEARCH<x>:TRIGger:A:LOGic:PAter:n:INPut:REF<x>	Sets or returns the Boolean logic criteria for the logic search
SEARCH:SEARCH<x>:TRIGger:A:LOGic:PAter:n:WHEn	Sets or returns the condition for generating a logic pattern search
SEARCH:SEARCH<x>:TRIGger:A:LOGic:PAter:n:WHEn:LESSLimit	Sets or returns the maximum time that the selected pattern may be true
SEARCH:SEARCH<x>:TRIGger:A:LOGic:PAter:n:WHEn:MORELimit	Sets or returns the minimum time that the selected pattern may be true
SEARCH:SEARCH<x>:TRIGger:A:LOGic:THReshold:CH<x>	Sets or returns the channel threshold level for an edge search
SEARCH:SEARCH<x>:TRIGger:A:LOGic:THReshold:MATH	Sets or returns the math waveform threshold level for edge search
SEARCH:SEARCH<x>:TRIGger:A:LOGic:THReshold:REF<x>	Sets or returns the reference waveform threshold level for edge search
SEARCH:SEARCH<x>:TRIGger:A:LOWerthreshold:CH<x>	Sets or returns the channel waveform lower threshold level for a transition search
SEARCH:SEARCH<x>:TRIGger:A:LOWerthreshold:MATH	Sets or returns the math waveform lower threshold level for a transition search
SEARCH:SEARCH<x>:TRIGger:A:LOWerthreshold:REF<x>	Sets or returns the reference waveform lower threshold level for a transition search
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:POLarity	Sets or returns the polarity for a pulse search
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:SOUrce	Sets or returns the source waveform for a pulse search
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:WHEn	Sets or returns the condition for generating a pulse width search
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:WIDth	Sets or returns the pulse width setting for a pulse width search
SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLarity	Sets or returns the polarity setting for a runt search
SEARCH:SEARCH<x>:TRIGger:A:RUNT:SOUrce	Sets or returns the source setting for a runt search
SEARCH:SEARCH<x>:TRIGger:A:RUNT:WHEn	sets or returns the condition setting for a runt search
SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDth	Sets or returns the width setting for a runt search
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCK:EDGE	Sets or returns the clock slope setting for a setup/hold search

**Search Commands, (cont.)**

<b>Command</b>	<b>Description</b>
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCK:SOUrce	Sets or returns the clock source setting for an setup/hold search
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCK:THReshold	Sets or returns the clock threshold setting for an setup/hold search
SEARCH:SEARCH<x>:TRIGger:A:SETHold:DATA:SOUrce	Sets or returns the data source setting for an setup/hold search
SEARCH:SEARCH<x>:TRIGger:A:SETHold:DATA:THReshold	Sets or returns the data threshold setting for an setup/hold search
SEARCH:SEARCH<x>:TRIGger:A:SETHold:HOLDTime	Sets or returns the hold time setting for an setup/hold search
SEARCH:SEARCH<x>:TRIGger:A:SETHold:SETTime	Sets or returns the setup time setting for an setup/hold search
SEARCH:SEARCH<x>:TRIGger:A:TRANSition :RISEFall :DELtAtime	Sets or returns the transition time setting for an transition search
SEARCH:SEARCH<x>:TRIGger:A:TRANSition :RISEFall :POLarity	Sets or returns the polarity setting for an transition search
SEARCH:SEARCH<x>:TRIGger:A:TRANSition :RISEFall :SOUrce	Sets or returns the source setting for an transition search
SEARCH:SEARCH<x>:TRIGger:A:TRANSition :RISEFall :WHEn	Sets or returns the condition setting for an transition search
SEARCH:SEARCH<x>:TRIGger:A:TYPE	Sets or returns the trigger type setting for a search
SEARCH:SEARCH<x>:TRIGger:A:UPPerthreshold:CH<x>	Sets or returns the channel waveform upper threshold level for a transition search
SEARCH:SEARCH<x>:TRIGger:A:UPPerthreshold:MATH	Sets or returns the math waveform upper threshold level for a transition search
SEARCH:SEARCH<x>:TRIGger:A:UPPerthreshold:REF<x>	Sets or returns the reference waveform upper threshold level for a transition search

## Status and Error Command Group

Use the commands in the Status and Error command Group to determine the status of the instrument and control events.

Several commands and queries used with the instrument are common to all devices on the GPIB bus. The IEEE Std 488.2-1987 defines these commands and queries. The common commands begin with an asterisk (\*) character.

Table 2-26: Status and Error Commands

Command	Description
ALLEv?	Returns all events and their messages
BUSY?	Returns instrument status
*CLS	Clears status
DESE	Sets or returns the bits in the Device Event Status Enable Register
*ESE	Sets or returns the bits in the Event Status Enable Register
*ESR?	Returns the contents of the Standard Event Status Register
EVENT?	Returns event code from the event queue
EVMsg?	Returns event code, message from the event queue
EVQty?	Return number of events in the event queue
*OPC	Generates the operation complete message in the standard event status register when all pending operations are finished Or returns "1" when all current operations are finished
*OPT?	Returns a list of options installed in the instrument
*PSC	Sets or returns the power on status flag
*PUD	Sets or returns a string of protected user data
*RST	Resets the instrument to factory default settings
*SRE	Sets or returns the bits in the Service Request Enable Register
*STB?	Returns the contents of the Status Byte Register
*WAI	Prevents the instrument from executing further commands until all pending operations finish

## Trigger Command Group

Use the commands in the Trigger Command Group to control all aspects of triggering for the instrument.

There are two triggers: A and B. Where appropriate, the command set has parallel constructions for each trigger.

You can set the A or B triggers to edge mode. Edge triggering lets you display a waveform at or near the point where the signal passes through a voltage level of your choosing.

You can also set A triggers to pulse, logic or video modes. With pulse triggering, the instrument triggers whenever it detects a pulse of a certain width or height. Logic triggering lets you logically combine the signals on one or more channels. The instrument then triggers when it detects a certain combination of signal levels. Video triggering enables you to trigger on the most common Standard Definition, High Definition, and custom video standards. You can trigger on all fields, all lines, or individual video lines.

**Table 2-27: Trigger Commands**

Command	Description
<a href="#">BUS?</a>	Returns the parameters for each bus
<a href="#">BUS:B&lt;x&gt;:CAN:BITRate</a>	Sets or returns the bit rate for the CAN bus
<a href="#">BUS:B&lt;x&gt;:CAN:PRObe</a>	Sets or returns the probing method used to probe the CAN bus
<a href="#">BUS:B&lt;x&gt;:CAN:SAMPLEpoint</a>	Sets or returns the sample point (in %) to sample during each bit period
<a href="#">BUS:B&lt;x&gt;:CAN:SOUrce</a>	Sets or returns the CAN data source
<a href="#">BUS:B&lt;x&gt;:DISPLAY:FORMAT</a>	Sets the display format for the numerical information in the bus waveform
<a href="#">BUS:B&lt;x&gt;:DISPLAY:TYPE</a>	Sets the display type for the bus
<a href="#">BUS:B&lt;x&gt;:I2C{CLOCK SCLK}:SOUrce</a>	Sets or returns the I2C SCLK source
<a href="#">BUS:B&lt;x&gt;:I2C{DATA SDATA}:SOUrce</a>	Sets or returns the I2C SDATA source
<a href="#">BUS:B&lt;x&gt;:POSition</a>	Sets or returns the position of the bus waveforms
<a href="#">BUS:B&lt;x&gt;:SPI{CLOCK SCLK}:POLARity</a>	Sets or returns the SPI SCLK polarity
<a href="#">BUS:B&lt;x&gt;:SPI{CLOCK SCLK}:SOUrce</a>	Sets or returns the SPI SCLK source
<a href="#">BUS:B&lt;x&gt;:SPI:DATA{IN MISO}:POLARity</a>	Sets or returns the SPI MISO polarity
<a href="#">BUS:B&lt;x&gt;:SPI:DATA{IN MISO}:SOUrce</a>	Sets or returns the SPI MISO source
<a href="#">BUS:B&lt;x&gt;:SPI:DATA{OUT MOSI}:POLARity</a>	Sets or returns the SPI MOSI polarity
<a href="#">BUS:B&lt;x&gt;:SPI:DATA{OUT MOSI}:SOUrce</a>	Sets or returns the SPI MOSI source
<a href="#">BUS:B&lt;x&gt;:SPI{SElect SS}:POLARity</a>	Sets or returns the SPI SS polarity
<a href="#">BUS:B&lt;x&gt;:SPI{SElect SS}:SOUrce</a>	Sets or returns the SPI SS source
<a href="#">BUS:B&lt;x&gt;:STATE</a>	Turns the bus on and off
<a href="#">BUS:B&lt;x&gt;:TYPE</a>	Sets or returns the bus type
<a href="#">BUS:THReshold:CH&lt;x&gt;</a>	Sets or returns the threshold for a channel
<a href="#">TRIGger</a>	Forces a trigger event to occur
<a href="#">TRIGger:A</a>	Sets A trigger level to 50% or returns current A trigger parameters

Table 2-27: Trigger Commands, (cont.)

Command	Description
TRIGger:A:BUS	Sets or queries the serial trigger type
TRIGger:A:BUS:B<x>:CAN:CONDition	Sets or returns the CAN condition
TRIGger:A:BUS:B<x>:CAN:DATA:DIRection	Sets or queries the CAN trigger condition to be valid on a READ, WRITE, or either
TRIGger:A:BUS:B<x>:CAN:DATA:QUALifier	Sets or returns the CAN data qualifier
TRIGger:A:BUS:B<x>:CAN:DATA:SIZE	Sets or returns the length of the data string in bytes to be used for CAN trigger
TRIGger:A:BUS:B<x>:CAN:DATA:VALue	Sets or returns the binary data string to be used for CAN trigger
TRIGger:A:BUS:B<x>:CAN:FRAMEtype	Sets or queries the CAN trigger frame type
TRIGger:A:BUS:B<x>:CAN{:IDentifier :ADDRess}:MODE	Sets or returns the CAN addressing mode
TRIGger:A:BUS:B<x>:CAN{:IDentifier :ADDRess}:VALue	Sets or queries the binary address string used for the CAN trigger
TRIGger:A:BUS:B<x>:I2C:ADDRess:MODE	Sets or queries the I2C address mode to 7 or 10-bit
TRIGger:A:BUS:B<x>:I2C:ADDRess:TYPE	Sets or returns the I2C address type to I2C special addresses
TRIGger:A:BUS:B<x>:I2C:ADDRess:VALue	Sets or queries the binary address string used for the I2C trigger
TRIGger:A:BUS:B<x>:I2C:CONDition	Sets or returns the trigger condition for I2C trigger
TRIGger:A:BUS:B<x>:I2C:DATA:DIRection	Sets or queries the I2C trigger condition valid on a READ, WRITE, or either
TRIGger:A:BUS:B<x>:I2C:DATA:SIZE	Sets or returns the length of the data string in bytes to be used for I2C trigger
TRIGger:A:BUS:B<x>:I2C:DATA:VALue	Sets or queries the binary data string used for I2C triggering
TRIGger:A:BUS:B<x>:SPI:CONDition	Sets or queries the trigger condition for SPI triggering
TRIGger:A:BUS:B<x>:SPI:DATA{:IN :MISO}:VALue	Sets or returns the binary data string to be used for SPI trigger
TRIGger:A:BUS:B<x>:SPI:DATA{:OUT :MOSI}:VALue	Sets or queries the binary data string used for the SPI trigger
TRIGger:A:BUS:B<x>:SPI:DATA:SIZE	Sets or returns the length of the data string in bytes to be used for SPI trigger
TRIGger:A:BUS:SOURce	Sets or queries the bus for a serial search
TRIGger:A:EDGE?	Returns the source, coupling and source for the A edge trigger
TRIGger:A:EDGE:COUPling	Sets or returns the type of coupling for the A edge trigger

Table 2-27: Trigger Commands, (cont.)

Command	Description
TRIGger:A:EDGE:SLOpe	Sets or returns the slope for the A edge trigger
TRIGger:A:EDGE:SOUrce	Sets or returns the source for the A edge trigger
TRIGger:A:HOLDoff?	Returns the A trigger holdoff parameters
TRIGger:A:HOLDoff:TIME	Sets or returns the A trigger holdoff time
TRIGger:A:LEVel	Sets or returns the level for the A trigger
TRIGger:A:LEVel:CH<x>	Specifies or returns the input settings for the specified trigger channel
TRIGger:A:LOGic?	Returns all A trigger logic settings
TRIGger:A:LOGic:CLAss	Sets or returns the type of A trigger logic
TRIGger:A:LOGic:FUNCTion	Sets or returns the logical combination of the input channels for the A logic trigger
TRIGger:A:LOGic:INPut?	Returns the A logic trigger input for channels 1-3.
TRIGger:A:LOGic:INPut:CH<x>	Specifies or returns the input settings for the specified logic trigger channel
TRIGger:A:LOGic:INPut:CLOCK:EDGE	Sets the polarity of the clock channel.
TRIGger:A:LOGic:INPut:CLOCK:SOUrce	Sets or returns the channel to use as the clock source
TRIGger:A:LOGic:PATtern?	Returns the conditions for generating an A logic pattern trigger
TRIGger:A:LOGic:PATtern:INPut:CH2	Sets or returns the A logic trigger input for channel 2
TRIGger:A:LOGic:PATtern:INPut:CH4	Sets or returns the A logic trigger input for channel 4
TRIGger:A:LOGic:PATtern:WHEn	Sets or returns the condition for generating the A logic pattern trigger
TRIGger:A:LOGic:PATtern:WHEn:LESSLimit	Sets or returns the maximum time that the selected pattern may be true and still generate an A logic pattern trigger
TRIGger:A:LOGic:PATtern:WHEn:MORELimit	Sets or returns the minimum time that the selected pattern may be true and still generate an A logic pattern trigger
TRIGger:A:SETHold?	Returns clock edge polarity, voltage threshold and source; data voltage threshold and source; and setup/hold times for violation triggering
TRIGger:A:SETHold:CLOCK?	Returns clock edge polarity, voltage threshold and source input for setup/hold triggering
TRIGger:A:LOGic:SETHold:CLOCK:EDGE	Sets or returns the clock edge polarity for setup and hold triggering

Table 2-27: Trigger Commands, (cont.)

Command	Description
TRIGger:A:LOGic:SETHold:CLOCK:SOURce	Sets or returns the clock source for the A logic trigger setup and hold input
TRIGger:A:LOGic:SETHold:CLOCK:THReshold	Sets or returns the clock voltage threshold for setup and hold trigger
TRIGger:A:SETHold:DATA?	Returns the voltage threshold and data source for the setup/hold trigger
TRIGger:A:LOGic:SETHold:DATA:SOURce	Sets or returns the data source for the setup and hold trigger
TRIGger:A:LOGic:SETHold:DATA:THReshold	Sets or returns the data voltage threshold for setup and hold trigger
TRIGger:A:LOGic:SETHold:HOLDTime	Sets or returns the hold time for the setup and hold violation triggering
TRIGger:A:LOGic:SETHold:SETTime	Sets or returns the setup time for setup and hold violation triggering
TRIGger:A:LOGic:THReshold?	Returns the threshold voltage for all channels in A logic trigger
TRIGger:A:LOGic:THReshold:CH<x>	Sets or returns the A logic trigger threshold voltage for the specified channel
TRIGger:A:LOWerthreshold:CH<x>	This command sets or returns the lower threshold for the channel selected
TRIGger:A:MODE	Sets or returns the A trigger mode
TRIGger:A:PULse?	Returns the A pulse trigger parameters
TRIGger:A:PULse:CLAss	Sets or returns the type of pulse on which to trigger
TRIGger:A:RUNT?	Returns the current A runt pulse trigger logic parameters
TRIGger:A:PULse:RUNT:HIGHLimit	Sets or returns the upper limit for the A pulse runt trigger
TRIGger:A:PULse:RUNT:LOWLimit	Sets or returns the lower limit for the A pulse runt trigger
TRIGger:A:PULse:RUNT:POLarity	Sets or returns the polarity for the A pulse runt trigger
TRIGger:A:PULse:RUNT:SOURce	Sets or returns the source for the A pulse trigger
TRIGger:A:PULse:RUNT:THReshold?	Returns the upper and lower thresholds for the A pulse runt trigger
TRIGger:A:PULse:RUNT:THReshold:HIGH	Sets or returns the upper limit for the A pulse runt trigger
TRIGger:A:PULse:RUNT:THReshold:LOW	Sets or returns the lower limit for the A pulse runt trigger
TRIGger:A:PULse:RUNT:WHEN	Sets or returns the type of pulse width the trigger checks for when it uncovers a runt

Table 2-27: Trigger Commands, (cont.)

Command	Description
TRIGger:A:PULse:RUNT:WIDTh	Sets or returns the minimum width for A pulse runt trigger
TRIGger:A{:TRANsition :RISEFall}?	Returns the delta time, polarity, and both upper and lower threshold limits for the transition time trigger
TRIGger:A:PULse{:TRANsition :SLEWRate :DELtAtime	Sets or returns the delta time used in calculating the transition value
TRIGger:A:PULse{:TRANsition :SLEWRate :HIGHLimit	Sets or returns the upper transition trigger threshold
TRIGger:A:PULse{:TRANsition :SLEWRate :LOWLimit	Sets or returns the lower transition trigger threshold
TRIGger:A:PULse{:TRANsition :SLEWRate :POLarity	Sets or returns the polarity for the A pulse transition trigger
TRIGger:A:PULse{:TRANsition :SLEWRate :SOURce	Sets or returns the source for transition trigger
TRIGger:A:PULse{:TRANsition :SLEWRate :THReshold?	Returns the upper and lower thresholds for the transition trigger
TRIGger:A:PULse{:TRANsition :SLEWRate :THReshold:HIGH	Sets or returns the upper transition trigger threshold
TRIGger:A:PULse{:TRANsition :SLEWRate :THReshold:LOW	Sets or returns the lower transition trigger threshold
TRIGger:A:PULse{:TRANsition :SLEWRate :WHEn	Sets or returns the relationship of delta time to transitioning signal
TRIGger:A:PULSEWIDth?	Returns the trigger A pulse width parameters
TRIGger:A:PULse:WIDth:HIGHLimit	Sets or returns the upper limit for the A pulse width trigger
TRIGger:A:PULse:WIDth:LEVel	Sets or returns the threshold for the pulse width trigger
TRIGger:A:PULse:WIDth:LOWLimit	Sets or returns the lower limit for the A pulse width trigger
TRIGger:A:PULse:WIDth:POLarity	Sets or returns the polarity for the A pulse width trigger
TRIGger:A:PULse:WIDth:SOURce	Sets or returns the source for the pulse width trigger
TRIGger:A:PULse:WIDth:WHEn	Sets or returns the criteria for width specification of pulse width trigger events
TRIGger:A:PULse:WIDth:WIDth	Sets or returns the width setting for the pulse width trigger
TRIGger:A:PULSEWIDth:POLarity	Sets or returns the polarity for the A pulse width trigger
TRIGger:A:PULSEWIDth:SOURce	Sets or returns the source for the pulse width trigger



Table 2-27: Trigger Commands, (cont.)

Command	Description
TRIGger:A:PULSEWidth:WHEn	Sets or returns the criteria for width specification of pulse width trigger events
TRIGger:A:PULSEWidth:WIDth	Sets or returns the width setting for the pulse width trigger
TRIGger:A:RUNT:POLarity	Sets or returns the polarity for the A pulse runt trigger
TRIGger:A:RUNT:SOUrce	Sets or returns the source for the A pulse trigger
TRIGger:A:RUNT:WHEn	Sets or returns the type of pulse width the trigger checks for when it uncovers a runt
TRIGger:A:RUNT:WIDth	Sets or returns the minimum width for A pulse runt trigger
TRIGger:A:SETHold:CLOCK:EDGE	Sets or returns the clock edge polarity for setup and hold triggering
TRIGger:A:SETHold:CLOCK:SOUrce	Sets or returns the clock source for the A logic trigger setup and hold input
TRIGger:A:SETHold:CLOCK:THReshold	Sets or returns the clock voltage threshold for setup and hold trigger
TRIGger:A:SETHold:DATA:SOUrce	Sets or returns the data source for the setup and hold trigger
TRIGger:A:SETHold:DATA:THReshold	Sets or returns the data voltage threshold for setup and hold trigger
TRIGger:A:SETHold:HOLDTime	Sets or returns the hold time for the setup and hold violation triggering
TRIGger:A:SETHold:SETTime	Sets or returns the setup time for setup and hold violation triggering
TRIGger:A{:TRANSition :RISEFall}:DELTAtime	Sets or returns the delta time used in calculating the transition value
TRIGger:A{:TRANSition :RISEFall}:POLarity	Sets or returns the polarity for the A pulse transition trigger
TRIGger:A{:TRANSition :RISEFall}:SOUrce	Sets or returns the source for transition trigger
TRIGger:A{:TRANSition :RISEFall}:WHEn	Sets or returns the relationship of delta time to transitioning signal
TRIGger:A:TYPE	Sets or queries the type of A trigger
TRIGger:A:UPPerthreshold:CH<x>	Sets the upper threshold for the channel selected
TRIGger:A:VIDeo?	Returns the video parameters for the A trigger
TRIGger:A:VIDeo:CUSTom?	Returns the custom video parameters for the A trigger

Table 2-27: Trigger Commands, (cont.)

Command	Description
TRIGger:A:VIDeo:CUSTom{:FORMat :TYPE}	Sets or queries the video trigger format
TRIGger:A:VIDeo:CUSTom:SCAN	Sets or returns the video scan rate
TRIGger:A:VIDeo:HDTv:FORMat	Sets or queries the analog HDTV video signal format
TRIGger:A:VIDeo:HOLDoff:FIELD	Sets or returns the trigger holdoff in video field units
TRIGger:A:VIDeo:LINE	Sets or returns the trigger delay as a number of video lines
TRIGger:A:VIDeo:POLarity	Sets or returns the polarity of the video trigger
TRIGger:A:VIDeo:SOUrce	Sets or returns the polarity of the video trigger
TRIGger:A:VIDeo:STANdard	Sets or returns the video standard
TRIGger:A:VIDeo{:SYNC :FIELD}	Sets or returns the video field trigger
TRIGger:B	Sets the B trigger level to 50% or returns the B trigger parameters
TRIGger:B:BY	Sets or returns B trigger time or event qualifiers
TRIGger:B:EDGE?	Returns B trigger edge type parameters
TRIGger:B:EDGE:COUPling	Sets or returns the type of B trigger coupling
TRIGger:B:EDGE:SLOpe	Sets or returns the B edge trigger slope
TRIGger:B:EDGE:SOUrce	Sets or returns the B edge trigger source
TRIGger:B:EVENTS?	Returns the current B trigger events parameter
TRIGger:B:EVENTS:COUNt	Sets or returns the number of events that must occur before the B trigger occurs
TRIGger:B:LEVel	Sets or returns the level for the B trigger
TRIGger:B:LEVel:CH<x>	Sets or returns the level for the B trigger for a specific channel
TRIGger:B:LOWerthreshold:CH<x>	Sets or returns the B trigger lower threshold for the channel selected
TRIGger:B:STATE	Returns the current state of the B trigger
TRIGger:B:TIME	Sets or returns the B trigger delay time
TRIGger:B:TYPE	Sets or returns the type of B trigger

Table 2-27: Trigger Commands, (cont.)

Command	Description
TRIGger:B:UPPerthreshold:CH<x>	Sets or returns the B trigger upper threshold for the channel selected
TRIGger:STATE?	Returns the current state of the triggering system

## Vertical Command Group

Use the commands in the Vertical Command Group to control the vertical setup of all live (channel) waveforms for acquisition and to control the display of channel, reference, and math waveforms.

You may replace VOLts with SCALE in the vertical commands. This provides program compatibility with earlier models of Tektronix instruments.

Table 2-28: Vertical Commands

Command	Description
CH<x>?	Returns vertical parameters for the specified channel
CH<x>:BANdwidth	Sets or returns the bandwidth of the specified channel
CH<x>:COUPling	Sets or returns the coupling setting for the specified channel
CH<x>:DESKew	Sets or returns the deskew time for the specified channel
CH<x>:INVert	Sets or returns the invert function for the specified channel
CH<x>:OFFSet	Sets or returns the channel offset
CH<x>:POSition	Sets or returns the channel vertical position
CH<x>:PRObe?	Returns the gain, resistance, units, and ID of the probe that is attached to the specified channel
CH<x>:PRObe:AUTOZero	Executes the probe auto-zero operation
CH<x>:PRObe:DEGAUss	Executes a probe degauss operation
CH<x>:PRObe:DEGAUss:STATE?	Returns the probe degauss state
CH<x>:PRObe:FORCERange	
CH<x>:PRObe:GAIN	Returns the gain of the probe that is attached to the specified channel
CH<x>:PRObe:ID?	Returns the type and serial number of the probe that is attached to the specified channel

Table 2-28: Vertical Commands, (cont.)

Command	Description
CH<x>:PRObe:ID:SERnumber?	Returns the serial number of the probe that is attached to the specified channel
CH<x>:PRObe:ID:TYPE?	Returns the type of probe that is attached to the specified channel
CH<x>:PRObe:RESistance?	Returns the resistance of the probe that is attached to the specified channel
CH<x>:PRObe:SIGnal	
CH<x>:PRObe:UNIts?	Returns the units of the probe that is attached to the specified channel
CH<x>:SCAle	Sets or returns the vertical scale of the specified channel
CH<x>:TERmination	Sets channel input termination
CH<x>:YUNits	Sets or returns vertical units
	Returns digital channel parameters
	Sets or returns the digital channel vertical position
	Sets or returns the threshold for the digital channel
REF<x>:DATE?	Returns the date that a reference waveform was stored
REF<x>:HORizontal:POSition	Sets or returns the horizontal position of the specified reference waveform in percent of the waveform that is displayed to the right of the center vertical graticule
REF<x>:HORizontal:SCAle	Sets or returns the horizontal scale for a reference waveform
REF<x>:TIme?	Returns the time that a reference waveform was stored
REF<x>:VERTical:POSition	Sets or returns the vertical position of the specified reference waveform
REF<x>:VERTical:SCAle	Sets or returns the reference waveform vertical scale in vertical units/div
SElect?	Returns information on which waveforms are on or off and which waveform is selected.
SElect:BUS<x>	Turns on the specified bus waveform Or returns whether the specified bus channel is on or off
SElect:CH<x>	Turns on the specified waveform Or returns whether the specified channel is on or off

Table 2-28: Vertical Commands, (cont.)

Command	Description
	Turns on the specified digital waveform Or returns whether the specified digital channel is on or off
SElect:CONTROL	Sets or returns the waveform controlled by the front panel
SElect:MATH[1]	Turns on the math waveform Or returns whether the math waveform is on or off
SElect:REF<x>	Turns on the specified reference waveform Or returns whether the specified reference waveform is on or off

## Waveform Transfer Command Group

Use the commands in the Waveform Transfer Command Group to transfer waveform data points to and from the instrument. Waveform data points are a collection of values that define a waveform. One data value usually represents one data point in the waveform record. When working with envelope waveforms, each data value is either the minimum or maximum of a min/max pair.

Before you transfer waveform data, you must specify the data format, record length, and waveform source.

### Data Formats

Acquired waveform data uses eight or more bits to represent each data point. The number of bits used depends on the acquisition mode specified when you acquired the data. Data acquired in SAMple or ENVELOpe mode uses eight bits per waveform data point. Data acquired in AVERage mode uses up to 14 bits per point.

The instrument can transfer waveform data in either ASCII or binary format. You specify the format with the DATA:ENCdg command. The instrument uses signed, 4 byte integers and floating point values; it does not support unsigned floating point values.

**ASCII Data.** ASCII data is represented by signed integer or floating point values. An example ASCII waveform data string may look like this:

```
CURVE<space>-110,-109,-110,-110,-109,-107,
-109,-107,-106,-105,-103,-100,-97,-90,-84,-80
```

Use ASCII to obtain more readable and easier to format output than binary. However, ASCII may require more bytes to send the same values than it does with binary. This may reduce transmission speeds.

**Binary Data.** **Binary data** can be represented by signed integer or floating point values. The range of the values depends on the byte width specified. When the byte width is one, signed integer data ranges from -128 to 127, and positive integer values range from 0 to 255. When the byte width is two, the values range from -32768 to 32767. When a MATH (or REF that came with a MATH) is used, 32 bit floating point values are used that are four bytes in width.

The defined binary formats specify the order in which the bytes are transferred. The following are the four binary formats:

- RIBinary specifies signed integer data-point representation with the most significant byte transferred first.
- SRIBinary is the same as RIBinary except that the byte order is swapped, meaning that the least significant byte is transferred first. This format is useful when transferring data to PCs.
- RFBinary specifies floating point data-point representation with the most significant byte transferred first.
- SRFBinary is the same as RFBinary except that the byte order is swapped, meaning that the least significant byte is transferred first. This format is useful when transferring data to PCs.

## Waveform Data and Record Lengths

You can transfer multiple points for each waveform record. You can transfer a portion of the waveform or you can transfer the entire record. You can use the DATA:START and DATA:STOP commands to specify the first and last data points of the waveform record.

When transferring data into the instrument, you must first specify the record length of the destination waveform record. You do this with the WFMInpre:NR\_Pt command. Next, specify the first data point within the waveform record. For example, when you set DATA:START to 1, data points will be stored starting with the first point in the record. The instrument will ignore the value set by DATA:STOP when reading in data. It will stop reading data when there is no more data to read or when it has reached the specified record length.

When transferring data from the instrument, you must specify the first and last data points in the waveform record. Setting DATA:START to 1 and DATA:STOP to the record length will always return the entire waveform.

## Waveform Data Locations and Memory Allocation

The DATA:SOURce command specifies the waveform source when transferring a waveform from the instrument. You can only transfer one waveform at a time. Waveforms sent to the instrument are always stored in one of the four reference memory locations. Use the DATA:DESTination command to specify a reference memory location.

**Waveform Preamble**

Each waveform that you transfer has an associated waveform preamble that contains information such as the horizontal scale, the vertical scale, and other settings in effect when the waveform was created. Refer to the individual WFMInpre and WFMOutpre commands for more information.

**Scaling Waveform Data**

Once you transfer the waveform data to the controller, you can convert the data points into voltage values for analysis using information from the waveform preamble.

**Transferring Waveform Data from the Instrument**

You can transfer waveforms from the instrument to an external controller using the following sequence:

1. Select the waveform source(s) using DATA:SOURce.
2. Specify the waveform data format using DATA:ENCdg.
3. Specify the number of bytes per data point using WFMOutpre:BYT\_Nr.

---

**NOTE.** *MATH waveforms (and REF waveforms that came from a MATH) are always set to four bytes.*

---

4. Specify the portion of the waveform that you want to transfer using DATA:STARt and DATA:STOP.
5. Transfer waveform preamble information using WFMOutpre.
6. Transfer waveform data from the instrument using CURVe.

**Transferring Waveform Data to the Instrument**

1. Specify waveform reference memory using DATA:DESTination.
2. Set WFMInpre:NR\_Pt to equal the number of data points to be sent.
3. Specify the waveform data format using WFMInpre:ENCdg.
4. Specify the number of bytes per data point using WFMInpre:BYT\_Nr.
5. Specify first data point in the waveform record using DATA:STARt.
6. Transfer waveform preamble information using WFMInpre.

---

**NOTE.** *FastAcq waveforms cannot be returned to the instrument via the CURVe command.*

---

Table 2-29: Waveform Transfer Commands

Command	Description
<a href="#">CURVe</a>	The command format transfers waveform data to the instrument (reference memory location specified by DATA:DESTination)  The query format transfers waveform data from instrument specified by the DATA:SOURce command
<a href="#">DATA</a>	Sets the format and location of the waveform data that is transferred with the CURVe Command  Or returns the format and location of the waveform data that is transferred with the CURVe? command
<a href="#">DATA:DESTination</a>	Sets or returns the reference memory location for storing waveform data sent to the instrument
<a href="#">DATA:ENCdg</a>	Sets or returns the format of outgoing waveform data
<a href="#">DATA:SOURce</a>	Sets or returns the location of waveform data transferred from the instrument
<a href="#">DATA:START</a>	Sets or returns the starting point in waveform transfer
<a href="#">DATA:STOP</a>	Sets or returns the ending data point in waveform transfer
<a href="#">WAVFrm?</a>	Returns a branch query containing waveform data in either binary or ASCII format, waveform formatting data, and the location of the waveform data source
<a href="#">WFMInpre?</a>	Returns the waveform formatting specification to be applied to the next incoming CURVE command data
<a href="#">WFMInpre:BIT_Nr</a>	Sets or returns the number of bits per binary waveform point
<a href="#">WFMInpre:BN_Fmt</a>	Sets or returns the format of binary data for the incoming waveform
<a href="#">WFMInpre:BYT_Nr</a>	Sets or returns the binary field data width for the first ordered waveform
<a href="#">WFMInpre:BYT_Or</a>	Sets or returns the byte order of waveform points
<a href="#">WFMInpre:ENCdg</a>	Sets or returns the type of encoding for incoming waveform data
<a href="#">WFMInpre:NR_Pt</a>	Sets or returns the number of points in the transmitted waveform record



Table 2-29: Waveform Transfer Commands, (cont.)

Command	Description
WFMinpre:PT_Fmt	Sets or returns the point format of incoming waveform data
WFMinpre:PT_Off	Sets or returns the trigger point within the incoming waveform data
WFMinpre:WFId	Provided to allow a waveform extracted from the instrument to be more easily sent back
WFMinpre:XINcr	Sets or returns the horizontal sampling interval between incoming waveform points
WFMinpre:XUNit	Sets or returns the horizontal units of the incoming waveform
WFMinpre:XZEro	Sets or returns the (sub-sample) time between the trigger sample and the occurrence of the actual incoming waveform trigger
WFMinpre:YMUlt	Sets or returns the vertical scale factor, per digitizing level, of the incoming waveform points
WFMinpre:YOff	Sets or returns the vertical position of the incoming waveform in digitizing levels
WFMinpre:YUNit	Sets or returns the vertical units of the incoming waveform
WFMinpre:YZEro	Sets or returns the offset of the incoming waveform
WFMOupre?	Returns the waveform formatting data for the waveform specified by the DATA:SOURCE command
WFMOupre:BIT_Nr	Sets or returns the number of bits per waveform point that outgoing waveforms contain
WFMOupre:BN_Fmt	Sets or returns the format of binary data for the waveform
WFMOupre:BYT_Nr	Sets or returns the data width for the waveform
WFMOupre:BYT_Or	Sets or returns the byte order of waveform points
WFMOupre:ENCdg	Sets or returns the type of encoding for outgoing waveforms
WFMOupre:NR_Pt?	Returns the number of points for the waveform transmitted in response to a CURVe? query
WFMOupre:PT_Fmt?	Returns the point format for the waveform
WFMOupre:PT_Off?	Returns the trigger point relative to DATA:START for the waveform

**Table 2-29: Waveform Transfer Commands, (cont.)**

Command	Description
WFMOutpre:PT_Order?	Returns whether the source waveform is DPO
WFMOutpre:WFIId?	Returns a string describing the acquisition parameters for the waveform
WFMOutpre:XINcr?	Returns the horizontal sampling interval
WFMOutpre:XUNit?	Returns the horizontal units for the waveform
WFMOutpre:XZEro?	Returns the (sub-sample) time between the trigger sample and the occurrence of the actual waveform trigger
WFMOutpre:YMULT?	Returns the vertical scale factor per digitizing level for the waveform
WFMOutpre:YOFf?	Returns the vertical offset in digitizing levels for the waveform
WFMOutpre:YUNit?	Returns the vertical units for the waveform
WFMOutpre:YZEro?	Returns the vertical offset for the waveform

## Zoom Command Group

Zoom commands let you expand and position the waveform display horizontally and vertically, without changing the time base or vertical settings.

You can specify up to four zoom areas. You can also specify a zoomed waveform as a zoom source, which enables you to zoom in on a zoomed waveform.

**Table 2-30: Zoom Commands**

Command	Description
ZOOM?	Returns the current vertical and horizontal positioning and scaling of the display
ZOOM:GRAticule:SIZE?	Sets or returns the size of the zoom graticule
ZOOM:GRAticule:SPLit?	Sets or returns the sizes of the acquisition and zoom windows when Zoom is selected.
ZOOM{:MODEj:STATE}	Sets or returns the zoom mode
ZOOM:ZOOM<x>?	Sets or returns zoom factors for the specified zoom
ZOOM:ZOOM<x>:FACtor?	Returns the zoom factor of a particular zoom box
ZOOM:ZOOM<x>:HORizontal:POSition	Sets or returns the horizontal zoom position for the specified waveform in the specified zoom
ZOOM:ZOOM<x>:HORizontal:SCAle	Sets or returns the horizontal zoom scale of the specified waveform in the specified zoom

Table 2-30: Zoom Commands, (cont.)

Command	Description
<a href="#">ZOOM:ZOOM&lt;x&gt;:POSition</a>	Sets or returns the horizontal zoom position for the specified waveform in the specified zoom
<a href="#">ZOOM:ZOOM&lt;x&gt;:SCALE</a>	Sets or returns the horizontal zoom scale of the specified waveform in the specified zoom
<a href="#">ZOOM:ZOOM&lt;x&gt;:STATE</a>	Specifies or returns a trace as zoomed, on or off



---

# Commands Listed in Alphabetical Order

## ACQuire? (Query Only)

This query-only command returns the following current acquisition parameters:

- Stop after
- Acquisition state
- Mode
- Number of envelopes
- Number of averages
- Repetitive signals

**Group** Acquisition

**Syntax** ACQuire?

**Related Commands** [ACQuire:MODE](#), [ACQuire:NUMACq?](#), [ACQuire:NUMAVg](#),  
[ACQuire:NUMEnv](#), [ACQuire:STOPAfter](#)

**Examples** ACQUIRE? might return the following string for the current acquisition parameters:  
  
:ACQUIRE:STOPAFTER RUNSTOP;STATE 1;MODE SAMPLE;NUMENV  
10;NUMAVG 16;NUMSAMPLES 16000;SAMPLINGMODE IT

## ACQuire:MAXSamplerate? (Query Only)

Returns the maximum real-time sample rate which varies from model to model.

**Group** Acquisition

**Syntax** ACQuire:MAXSamplerate?

**Examples** ACQUIRE:MAXSAMPLERATE? might return 2.5e9 in a DPO4034 indicating the maximum real-time sample rate is 2.5GS/s.

## ACQuire:MODE

This command sets or returns the acquisition mode of the instrument. This affects all live waveforms. This command is equivalent to selecting Horizontal/Acquisition from the Horiz/Acq menu, and then choosing the desired mode from the Acquisition Mode group box.

Waveforms are the displayed data point values taken from acquisition intervals. Each acquisition interval represents a time duration set by the horizontal scale (time per division). The instrument sampling system always samples at the maximum rate and so an acquisition interval may include more than one sample.

The acquisition mode (which you set using this ACQuire:MODE command) determines how the final value of the acquisition interval is generated from the many data samples.

**Group** Acquisition

**Syntax** ACQuire:MODE {SAMPle|PEAKdetect|HIRes|AVERage|ENVELOpe}  
ACQuire:MODE?

**Related Commands** [ACQuire:NUMAVg](#), [ACQuire:NUMEnv](#), [CURVe](#)

**Arguments** **SAMPle** specifies that the displayed data point value is the first sampled value that is taken during the acquisition interval. In sample mode, all waveform data has 8 bits of precision. You can request 16 bit data with a [CURVe](#) query but the lower-order 8 bits of data will be zero. **SAMPle** is the default mode.

**PEAKdetect** specifies the display of high-low range of the samples taken from a single waveform acquisition. The high-low range is displayed as a vertical column that extends from the highest to the lowest value sampled during the acquisition interval. **PEAKdetect** mode can reveal the presence of aliasing or narrow spikes.

**HIRes** specifies Hi Res mode where the displayed data point value is the average of all the samples taken during the acquisition interval. This is a form of averaging, where the average comes from a single waveform acquisition. The number of samples taken during the acquisition interval determines the number of data values that compose the average.

**AVERage** specifies averaging mode, in which the resulting waveform shows an average of **SAMPle** data points from several separate waveform acquisitions. The instrument processes the number of waveforms you specify into the acquired waveform, creating a running exponential average of the input signal. The number of waveform acquisitions that go into making up the average waveform is set or queried using the [ACQuire:NUMAVg](#) command.

ENVELOPE specifies envelope mode, where the resulting waveform shows the PEAKdetect range of data points from several separate waveform acquisitions. The number of waveform acquisitions that go into making up the envelope waveform is set or queried using the ACQUIRE:NUMENV command.

**Examples** ACQUIRE:MODE ENVELOPE sets the acquisition mode to display a waveform that is an envelope of many individual waveform acquisitions.

ACQUIRE:MODE? might return :ACQUIRE:MODE AVERAGE indicating that the displayed waveform is the average of the specified number of waveform acquisitions.

## ACQUIRE:NUMACQ? (Query Only)

This query-only command returns the number of waveform acquisitions that have occurred since starting acquisition with the ACQUIRE:STATE RUN command. This value is reset to zero when any acquisition, horizontal, or vertical arguments that affect the waveform are changed. The maximum number of acquisitions that can be counted is  $2^{30}-1$ . The instrument stops counting when this number is reached. This is the same value that displays in the upper center of the screen when the acquisition system is stopped.

**Group** Acquisition

**Syntax** ACQUIRE:NUMACQ?

**Related Commands** [ACQUIRE:STATE](#)

**Examples** ACQUIRE:NUMACQ? might return :ACQUIRE:NUMACQ 350 indicating that 350 acquisitions have occurred since executing an ACQUIRE:STATE RUN command.

## ACQUIRE:NUMAVG

This command sets or returns the number of waveform acquisitions that make up an averaged waveform. Use the ACQUIRE:MODE command to enable the Average mode. Sending this command is equivalent to selecting Horizontal/Acquisition Setup from the Horiz/Acq menu, selecting the Acquisition tab, and choosing Average from the Acquisition Mode group box. Then enter the desired number of waveforms that will be used to make up an averaged waveform in the # of Wfms box.

<b>Group</b>	Acquisition
<b>Syntax</b>	ACQuire:NUMAVg <NR1> ACQuire:NUMAVg?
<b>Related Commands</b>	ACQuire:MODE
<b>Arguments</b>	NR1 is the number of waveform acquisitions to average. The range of values is from 2 to 512 in powers of two.
<b>Examples</b>	ACQUIRE:NUMAVG 16 specifies that 16 waveform averages will be performed before exponential averaging starts.  ACQUIRE:NUMAVG? might return :ACQUIRE:NUMAVG 64 indicating that there are 64 acquisitions specified for averaging.

## ACQuire:NUMEnv

This command sets or returns the number of waveform acquisitions that make up an envelope waveform. Sending this command is equivalent to setting the # of Wfms in the Acquisition Setup menu when Envelope Acquisition mode is selected.

<b>Group</b>	Acquisition
<b>Syntax</b>	ACQuire:NUMEnv INFInite ACQuire:NUMEnv?
<b>Arguments</b>	INFInite is the number of waveform acquisitions to be enveloped.
<b>Examples</b>	ACQUIRE:NUMENV? returns:ACQUIRE:NUMENV INFInite indicating that acquisitions are acquired infinitely for enveloped waveforms.

## ACQuire:STATE

This command starts or stops acquisitions. When state is set to ON or RUN, a new acquisition will be started. If the last acquisition was a single acquisition sequence, a new single sequence acquisition will be started. If the last acquisition was continuous, a new continuous acquisition will be started.



If RUN is issued in the middle of completing a single sequence acquisition (for example, averaging or enveloping), the acquisition sequence is restarted, and any accumulated data is discarded. Also, the instrument resets the number of acquisitions. If the RUN argument is issued while in continuous mode, a reset occurs and acquired data continues to acquire.

Sending this command is equivalent to pressing the front-panel RUN/STOP button.

**Group** Acquisition

**Syntax** ACQUIRE:STATE {OFF|ON|RUN|STOP|<NR1>}  
ACQUIRE:STATE?

**Related Commands** [ACQUIRE:STOPAfter](#)

**Arguments** OFF stops acquisitions.  
STOP stops acquisitions.  
ON starts acquisitions.  
RUN starts acquisitions.  
NR1 = 0 stops acquisitions; any other value starts acquisitions.

**Examples** ACQUIRE:STATE RUN starts the acquisition of waveform data and resets the count of the number of acquisitions.  
ACQUIRE:STATE? might return:ACQUIRE:STATE 0 indicating that the acquisition is stopped.

## ACQUIRE:STOPAfter

This command sets or returns whether the instrument continually acquires acquisitions or acquires a single sequence. This command is equivalent to pressing SINGLE from the front panel.

**Group** Acquisition

**Syntax** ACQUIRE:STOPAfter {RUNSTOP|SEQUENCE}  
ACQUIRE:STOPAfter?

**Related Commands**    [ACQuire:STATE](#)

**Arguments**    RUNSTOP specifies that the instrument will continually acquire data, if ACQuire:STATE is turned on.  
  
SEQuence specifies that the next acquisition will be a single-sequence acquisition.

**Examples**    ACQUIRE:STOPAFTER RUNSTOP sets the instrument to continually acquire data.  
  
ACQUIRE:STOPAFTER? might return: ACQUIRE:STOPAFTER SEQUENCE indicating that the next acquisition the instrument makes will be of the single-sequence type.

## ALias

This command sets or returns the state of alias functionality.

**Group**    Alias

**Syntax**    ALIAS {OFF|ON|<NR1>}  
ALIAS?

**Related Commands**    [ALias\[:STATE\]](#)

**Arguments**    OFF turns Alias expansion off. If a defined alias is sent when ALias:STATE is off, a command error (102) will be generated.  
  
ON turns Alias expansion on. When a defined alias is received, the specified command sequence is substituted for the alias and executed.  
  
NR1=0 disables Alias mode; any other value enables Alias mode.

**Examples**    ALIAS ON turns the alias feature on. When a defined alias is received, the specified command sequence is substituted for the alias and executed.  
  
ALIAS? might return :ALIAS:STATE 1 indicating that the alias feature is on.

## ALias:CATalog? (Query Only)

This query-only command returns a list of the currently defined alias labels, separated by commas. If no aliases are defined, the query returns the string "".

<b>Group</b>	Alias
<b>Syntax</b>	ALIAS:CATALOG?
<b>Examples</b>	ALIAS:CATALOG? might return the string :ALIAS:CATALOG "SETUP1", "TESTMENU1", "DEFAULT" showing that there are three aliases named SETUP1, TESTMENU1, and DEFAULT.

## Alias:DEFine

This command assigns a sequence of program messages to an alias label. These messages are then substituted for the alias whenever it is received as a command or query, provided that ALIAS:STATE has been turned on. The query form of this command returns the definitions of a selected alias.

---

**NOTE.** *Attempting to give two aliases the same name causes an error. To give a new alias the name of an existing alias, the existing alias must first be deleted.*

---

<b>Group</b>	Alias
<b>Syntax</b>	ALIAS:DEFine <QString><,>{<QString> <Block>} ALIAS:DEFine?
<b>Related Commands</b>	<a href="#">ALIAS[:STATE]</a>
<b>Arguments</b>	<p>The first &lt;QString&gt; is the alias label.</p> <p>This label cannot be a command name. Labels must start with a letter and can contain only letters, numbers, and underscores; other characters are not allowed. The label must be less than or equal to 12 characters.</p> <p>The second&lt;QString&gt; or &lt;Block&gt; is a complete sequence of program messages.</p> <p>The messages can contain only valid commands that must be separated by semicolons and must follow all rules for concatenating commands. The sequence must be less than or equal to 256 characters.</p>
<b>Examples</b>	ALIAS:DEFINE "ST1",":RECALL:SETUP 5;:AUTOSET EXECUTE;:SELECT:CH1 ON" defines an alias named "ST1" that sets up the instrument.

ALIAS:DEFINE? "ST1" returns:ALIAS:DEFINE "ST1",#246  
:RECALL:SETUP 5;:AUTOSET EXECUTE;:SELECT:CH1 ON

## Alias:DELEte (No Query Form)

This command removes a specified alias and is identical to Alias:DELEte:NAME. An error message is generated if the named alias does not exist.

**Group** Alias

**Syntax** ALIAS:DELEte <QString>

**Related Commands** \*ESR?, [Alias:DELEte:ALL](#)

**Arguments** <QString> is the name of the alias to be removed. Using Alias:DELEte without specifying an alias causes an execution error. <QString> must be a previously defined value.

**Examples** ALIAS:DELETE "SETUP1" deletes the alias named SETUP1.

## Alias:DELEte:ALL (No Query Form)

This command deletes all existing aliases.

**Group** Alias

**Syntax** ALIAS:DELEte:ALL

**Related Commands** [Alias:DELEte](#), [Alias:DELEte\[:NAME\]](#)

**Examples** ALIAS:DELETE:ALL deletes all existing aliases.

## Alias:DELEte[:NAME] (No Query Form)

Removes a specified alias. This command is identical to [Alias:DELEte](#)

**Group** Alias

<b>Syntax</b>	<code>ALIAS:DELETE[:NAME] &lt;QString&gt;</code>
<b>Arguments</b>	<code>&lt;QString&gt;</code> is the name of the alias to remove. Using <code>ALIAS:DELETE[:NAME]</code> without specifying an alias causes an execution error. <code>&lt;QString&gt;</code> must be an existing alias.
<b>Examples</b>	<code>ALIAS:DELETE[:NAME] "STARTUP"</code> deletes the alias named STARTUP.

## Alias[:STATE]

Turns aliases on or off. This command is identical to the [Alias](#) command.

<b>Group</b>	Alias
<b>Syntax</b>	<code>Alias[:STATE] {&lt;NR1&gt; OFF ON}</code> <code>Alias[:STATE]?</code>
<b>Arguments</b>	<code>&lt;NR1&gt; = 0</code> turns alias expansion off. If a defined alias is sent when <code>Alias:STATE</code> is OFF, a command error (102) is generated. <code>.9</code> or <code>9</code>  <code>&lt;NR1&gt; ≠ 0</code> turns alias expansion on. When a defined alias is received, the specified command sequence is substituted for the alias and executed.
<b>Examples</b>	<code>ALIAS[:STATE] OFF</code> turns the command alias feature off. <code>ALIAS[:STATE]?</code> returns 0 when the alias feature is off.

## ALLEV? (Query Only)

This query-only command prompts the instrument to return all events and their messages (delimited by commas), and removes the returned events from the Event Queue. Use the `*ESR?` query to enable the events to be returned. This command is similar to repeatedly sending `*EVMsg?` queries to the instrument.

<b>Group</b>	Status and Error
<b>Syntax</b>	<code>ALLEV?</code>
<b>Related Commands</b>	<a href="#">*ESR?</a> , <a href="#">EVMsg?</a>

**Examples** ALLEV? might return :ALLEV 2225,"Measurement error, No waveform to measure; "420,"Query UNTERMINATED;"

## AUTOSet (No Query Form)

This command (no query format) sets the vertical, horizontal, and trigger controls of the instrument to automatically acquire and display the selected waveform. (To autoset a video waveform, the video trigger must be set to video standard, not custom.) This is equivalent to pressing the front-panel AUTOSSET button. For a detailed description of autoset functionality, see *Autoset* in the index of the online help for your instrument.

**Group** Miscellaneous

**Syntax** AUTOSet {EXECute|UNDO|VIDeo|VLines|VFields}

**Related Commands** [DISplay:GRAticule](#)

**Arguments** EXECute autosets the displayed waveform; this is equivalent to pressing the front-panel AUTOSSET button. If the display is set to a PAL, MV, or IRE graticule, this argument forces the graticule display to full mode (frame, grid, and cross hair).

UNDO returns the instrument to the setting prior to executing an autoset.

VIDeo autosets the displayed waveform.

VLines autosets the displayed waveform.

VFields autosets the displayed waveform.

**Examples** AUTOSET VFields sets the instrument to trigger on all video fields.

## AUXout? (Query Only)

This query-only command returns the auxiliary output setup and is equivalent to selecting External Signals... from the Utilities menu, and then viewing the current settings for the AUX OUT Configuration.

**Group** Miscellaneous

**Syntax** AUXout?

**Related Commands**    [AUXout:SOUrce](#)

**Examples**    AUXOUT? might return :AUXOUT:SOURCE ATRIGGER;EDGE RISING indicating that the source at the BNC connector is set to the A trigger and the polarity is set to the rising edge of the trigger output signal.

## AUXout:EDGE

Sets or returns the polarity of the auxiliary output.

**Group**    Miscellaneous

**Syntax**    AUXout:EDGE {RISing|FALling}  
AUXout:EDGE?

## AUXout:SOUrce

This command sets or returns the trigger source at the BNC connection. This command is equivalent to selecting AUX OUT Configuration from the Utilities menu and then selecting the desired Configuration setting.

**Group**    Miscellaneous

**Syntax**    AUXout:SOUrce {ATRIGger|BTRIGger|MAIn|DELAyed}  
AUXout:SOUrce?

**Related Commands**    [AUXout?](#)

**Arguments**    ATRIGger or MAIn sets the source at the BNC connector to the main trigger.  
BTRIGger or DELAyed sets the source at the BNC connector to the delayed trigger.

**Examples**    AUXOUT:SOURCE? might return :AUXOUT:SOURCE ATRIGGER, indicating that the source at the BNC connector is set to the A trigger.

## BELI (No Query Form)

This command was previously used to beep an audio indicator and is provided for backward compatibility.

<b>Group</b>	Miscellaneous
<b>Syntax</b>	BEL1
<b>Examples</b>	BELL is accepted but does nothing.

## BUS? (Query Only)

This command returns the parameters for each bus.

<b>Conditions</b>	This command requires a DPO4AUTO or DPO4EMBD application module.
<b>Group</b>	Trigger
<b>Syntax</b>	BUS?

## BUS:B<x>:CAN:BITRate

This command sets or returns the bit rate for the CAN bus.

<b>Conditions</b>	This command requires a DPO4AUTO application module.
<b>Group</b>	Trigger
<b>Syntax</b>	BUS:B<x>:CAN:BITRate {RATE10K RATE20K RATE33K RATE50K RATE62K RATE83K RATE100K  RATE125K RATE250K RATE500K RATE800K RATE1M} BUS:B<x>:CAN:BITRate?
<b>Arguments</b>	RATE10K sets the bit rate to 10 kbps. RATE20K sets the bit rate to 20 kbps. RATE33K sets the bit rate to 33 kbps.



RATE50K sets the bit rate to 50 kbps.

RATE62K sets the bit rate to 62 kbps.

RATE83K sets the bit rate to 83 kbps.

RATE100K sets the bit rate to 100 kbps.

RATE125K sets the bit rate to 125 kbps.

RATE250K sets the bit rate to 250 kbps.

RATE500K sets the bit rate to 500 kbps.

RATE800K sets the bit rate to 800 kbps.

RATE1M sets the bit rate to 1 Mbps.

## **BUS:B<x>:CAN:PRObe**

This command sets or returns the probing method used to probe the CAN bus.

<b>Conditions</b>	This command requires a DPO4AUTO application module.
<b>Group</b>	Trigger
<b>Syntax</b>	<b>BUS:B&lt;x&gt;:CAN:PRObe</b> {CANH CANL RX TX DIFFerential} <b>BUS:B&lt;x&gt;:CAN:PRObe?</b>
<b>Arguments</b>	CANH specifies a single-ended bus signal, active high. CANL specifies a single-ended bus signal, active high. RX specifies the receive signal. TX specifies the transmit signal. DIFFerential specifies a differential signal.

## **BUS:B<x>:CAN:SAMPLEpoint**

This command sets or returns the sample point (in %) to sample during each bit period.

<b>Conditions</b>	This command requires a DPO4AUTO application module.
-------------------	--

**Group** Trigger

**Syntax** `BUS:B<x>:CAN:SAMPLEpoint <NR3>`  
`BUS:B<x>:CAN:SAMPLEpoint?`

**Arguments** <NR3> is the sample point in percent.

## **BUS:B<x>:CAN:SOURce**

This command sets or returns the CAN data source.

**Conditions** This command requires a DPO4AUTO application module.

**Group** Trigger

**Syntax** `BUS:B<x>:CAN:SOURce {CH1|CH2|CH3|CH4}`  
`BUS:B<x>:CAN:SOURce?`

**Arguments** CH<x> is the channel to use as the data source.

## **BUS:B<x>:DISPLAY:FORMAT**

Sets the display format for the numerical information in the bus waveform.

**Conditions** This command requires a DPO4AUTO or DPO4EMBD application module.

**Group** Trigger

**Syntax** `BUS:B<x>:DISPLAY:FORMAT {BINary|HEXadecimal}`  
`BUS:B<x>:DISPLAY:FORMAT?`

**Arguments** BINary specifies a binary data display.  
 HEXadecimal specifies a hexadecimal data display.

**BUS:B<x>:DISPLAY:TYPE**

Sets the display type for the bus. The bus can be set up to display the protocol information, the logic waveforms that comprise the bus, or both.

**Conditions** This command requires a DPO4AUTO or DPO4EMBD application module.

**Group** Trigger

**Syntax** `BUS:B<x>:DISPLAY:TYPE {BUS|WAVEforms|BOTH}`  
`BUS:B<x>:DISPLAY:TYPE?`

**Arguments** `BUS` specifies to display the bus waveforms only.  
`WAVEforms` specifies to display the logic waveforms only.  
`BOTH` specifies to display both the bus and logic waveforms.

**BUS:B<x>:I2C{:CLOCK|:SCLK}:SOURCE**

This command sets or returns the I2C SCLK source.

**Conditions** This command requires a DPO4EMBD application module.

**Group** Trigger

**Syntax** `BUS:B<x>:I2C{:CLOCK|:SCLK}:SOURCE {CH1|CH2|CH3|CH4}`  
`BUS:B<x>:I2C{:CLOCK|:SCLK}:SOURCE?`

**Arguments** `CH<x>` is the channel to use as the I2C SCLK source.

**BUS:B<x>:I2C{:DATA|:SDATA}:SOURCE**

This command sets or returns the I2C SDATA source.

**Conditions** This command requires a DPO4EMBD application module.

**Group** Trigger

**Syntax**     `BUS:B<x>:I2C{:DATA|:SDATA}:SOURCE {CH1|CH2|CH3|CH4}`  
`BUS:B<x>:I2C{:DATA|:SDATA}:SOURCE?`

**Arguments**     CH<x> is the channel to use as the I2C SDATA source.

## **BUS:B<x>:POSition**

This command sets or returns the position of the bus waveforms. This is just a display function.

**Conditions**     This command requires a DPO4AUTO or DPO4EMBD application module.

**Group**     Trigger

**Syntax**     `BUS:B<x>:POSition <NR3>`  
`BUS:B<x>:POSition?`

**Arguments**     <NR3> specifies the position.

## **BUS:B<x>:SPI{:CLOCK|:SCLK}:POLARity**

This command sets or returns the SPI SCLK polarity.

**Conditions**     This command requires a DPO4EMBD application module.

**Group**     Trigger

**Syntax**     `BUS:B<x>:SPI{:CLOCK|:SCLK}:POLARity {FALL|RISe}`  
`BUS:B<x>:SPI{:CLOCK|:SCLK}:POLARity?`

**Arguments**     FALL specifies the falling edge.  
RISe specifies the rising edge.

## **BUS:B<x>:SPI{:CLOCK|:SCLK}:SOUrce**

This command sets or returns the SPI SCLK source.

<b>Conditions</b>	This command requires a DPO4EMBD application module.
<b>Group</b>	Trigger
<b>Syntax</b>	<code>BUS:B&lt;x&gt;:SPI{:CLOCK :SCLK}:SOURce {CH1 CH2 CH3 CH4}</code> <code>BUS:B&lt;x&gt;:SPI{:CLOCK :SCLK}:SOURce?</code>
<b>Arguments</b>	CH<x> is the channel to use as the SPI SCLK source.

## **BUS:B<x>:SPI:DATA{:IN|:MISO}:POLARity**

This command sets or returns the SPI MISO polarity.

<b>Conditions</b>	This command requires a DPO4EMBD application module.
<b>Group</b>	Trigger
<b>Syntax</b>	<code>BUS:B&lt;x&gt;:SPI:DATA{:IN :MISO}:POLARity {LOW HIGH}</code> <code>BUS:B&lt;x&gt;:SPI:DATA{:IN :MISO}:POLARity?</code>
<b>Arguments</b>	LOW specifies an active low polarity. HIGH specifies an active high polarity.

## **BUS:B<x>:SPI:DATA{:IN|:MISO}:SOURce**

This command sets or returns the SPI MISO source.

<b>Conditions</b>	This command requires a DPO4EMBD application module.
<b>Group</b>	Trigger
<b>Syntax</b>	<code>BUS:B&lt;x&gt;:SPI:DATA{:IN :MISO}:SOURce {CH1 CH2 CH3 CH4 OFF}</code> <code>BUS:B&lt;x&gt;:SPI:DATA{:IN :MISO}:SOURce?</code>
<b>Arguments</b>	CH<x> is the channel to use as the SPI MISO source. OFF specifies no source.

## **BUS:B<x>:SPI:DATA{:OUT|:MOSI}:POLARity**

This command sets or returns the SPI MOSI polarity.

**Conditions** This command requires a DPO4EMBD application module.

**Group** Trigger

**Syntax** `BUS:B<x>:SPI:DATA{:OUT|:MOSI}:POLARity {LOW|HIGH}`  
`BUS:B<x>:SPI:DATA{:OUT|:MOSI}:POLARity?`

**Arguments** LOW specifies an active low polarity.  
HIGH specifies an active high polarity.

## **BUS:B<x>:SPI:DATA{:OUT|:MOSI}:SOUrce**

This command sets or returns the SPI MOSI source.

**Conditions** This command requires a DPO4EMBD application module.

**Group** Trigger

**Syntax** `BUS:B<x>:SPI:DATA{:OUT|:MOSI}:SOUrce {CH1|CH2|CH3|CH4|OFF}`  
`BUS:B<x>:SPI:DATA{:OUT|:MOSI}:SOUrce?`

**Arguments** CH<x> is the channel to use as the SPI MOSI source.  
OFF specifies no source.

## **BUS:B<x>:SPI{:SElect|:SS}:POLARity**

This command sets or returns the SPI SS polarity.

**Conditions** This command requires a DPO4EMBD application module.

**Group** Trigger

**Syntax**     `BUS:B<x>:SPI{:SElect|:SS}:POLARity {LOW|HIGH}`  
`BUS:B<x>:SPI{:SElect|:SS}:POLARity?`

**Arguments**     LOW specifies an active low polarity.  
                    HIGH specifies an active high polarity.

## **BUS:B<x>:SPI{:SElect|:SS}:SOUrce**

This command sets or returns the SPI SS source.

**Conditions**     This command requires a DPO4EMBD application module.

**Group**          Trigger

**Syntax**        `BUS:B<x>:SPI{:SElect|:SS}:SOUrce {CH1|CH2|CH3|CH4}`  
`BUS:B<x>:SPI{:SElect|:SS}:SOUrce?`

**Arguments**     CH<x> is the channel to use as the SPI SS source.  
                    OFF specifies no source.

## **BUS:B<x>:STATE**

This command turns the bus on and off.

**Conditions**     This command requires a DPO4AUTO or DPO4EMBD application module.

**Group**          Trigger

**Syntax**        `BUS:B<x>:STATE {<NR1>|OFF|ON}`  
`BUS:B<x>:STATE?`

**Arguments**     ON or <NR1> ≠ 0 turns on the bus state.  
                    OFF or <NR1> = 0 turns off the bus state.

## BUS:B<x>:TYPE

This command sets or returns the bus type: I2C, SPI or CAN.

**Conditions** This command requires a DPO4AUTO or DPO4EMBD application module.

**Group** Trigger

**Syntax** BUS:B<x>:TYPE {I2C|SPI|CAN}  
BUS:B<x>:TYPE?

**Arguments** I2C specifies the Inter-IC bus.  
SPI specifies the Serial Peripheral Interface bus.  
CAN specifies the Controller Area Network bus.

## BUS:THReshold:CH<x>

This command sets or returns the threshold for a channel. This setting applies to all trigger types that use the channel.

**Conditions** This command requires a DPO4AUTO or DPO4EMBD application module.

**Group** Trigger

**Syntax** BUS:THReshold:CH<x> {<NR3>|ECL|TTL}  
BUS:THReshold:CH<x>?

**Arguments** <NR3> specifies the threshold level, in volts.  
ECL specifies a –1.3 V threshold level.  
TTL specifies a 1.4 V threshold level.

## BUSY? (Query Only)

This query-only command returns the status of the instrument. This command allows you to synchronize the operation of the instrument with your application program.



<b>Group</b>	Status and Error								
<b>Syntax</b>	BUSY?								
<b>Related Commands</b>	<a href="#">*OPC</a> , <a href="#">*WAI</a>								
<b>Returns</b>	<p>&lt;NR1&gt; = 0 means that the instrument is not busy processing a command whose execution time is extensive.</p> <p>&lt;NR1&gt; = 1 means that the instrument is busy processing one of the commands listed in the table below.</p> <p><b>Commands that affect BUSY? response</b></p> <table> <tr> <th>Operation</th><th>Command</th></tr> <tr> <td>Single sequence acquisition</td><td><a href="#">ACQUIRE:STATE</a> ON or <a href="#">ACQUIRE:STATE</a> RUN or <a href="#">ACQUIRE:STATE1</a> (when <a href="#">ACQUIRE:STOPAfter</a> is set to SEQUENCE)</td></tr> <tr> <td>Hard copy operation</td><td><a href="#">HARDCopy</a> STArt</td></tr> <tr> <td>Calibration step</td><td>Refer to the optional instrument Service Manual.</td></tr> </table>	Operation	Command	Single sequence acquisition	<a href="#">ACQUIRE:STATE</a> ON or <a href="#">ACQUIRE:STATE</a> RUN or <a href="#">ACQUIRE:STATE1</a> (when <a href="#">ACQUIRE:STOPAfter</a> is set to SEQUENCE)	Hard copy operation	<a href="#">HARDCopy</a> STArt	Calibration step	Refer to the optional instrument Service Manual.
Operation	Command								
Single sequence acquisition	<a href="#">ACQUIRE:STATE</a> ON or <a href="#">ACQUIRE:STATE</a> RUN or <a href="#">ACQUIRE:STATE1</a> (when <a href="#">ACQUIRE:STOPAfter</a> is set to SEQUENCE)								
Hard copy operation	<a href="#">HARDCopy</a> STArt								
Calibration step	Refer to the optional instrument Service Manual.								
<b>Examples</b>	BUSY? might return :BUSY 1 indicating that the instrument is currently busy.								

## CAL? (Query Only)

This query-only command returns the status of instrument calibration, including internal signal path calibration, factory calibration, and probe calibration. This query also returns the time until notification when the next factory calibration is due. Refer to the service manual for factory calibration command descriptions.

<b>Group</b>	Calibration and Diagnostic
<b>Syntax</b>	CAL?
<b>Related Commands</b>	<a href="#">CALibrate:RESults:SPC?</a>
<b>Examples</b>	<p>CAL? might return :CALIBRATE:INTERNAL:STATUS PASS;:CALIBRATE:TEMPERATURE 32;FACTORY:FANCONTROL 2;STATUS PASS;STEPSTATUS FAIL;STEPSTIMULUS</p>

```
"0.0,0,0,0,END";NOTIFY:HOURS 10500;YEARS 5.5000;DUE
DUE;:CALIBRATE:RESULTS:SPC PASS;FACTORY PASS;FACTORY:VOLTAGE
PASS;FREQUENCY PASS;TRIGGER PASS;:CALIBRATE:PROBESATE:CH1
1;CH2 1;CH3 1;CH4 1
```

## \*CAL? (Query Only)

Runs an internal self-calibration and returns the oscilloscope calibration status.

---

**NOTE.** *Disconnect or otherwise remove all input signals prior to starting self-calibration. The self-calibration can take several minutes to complete.*

---

*No other commands are executed until calibration is complete.*

---

<b>Group</b>	Calibration and Diagnostic
<b>Syntax</b>	*CAL?
<b>Returns</b>	<NR1> = 1 indicates that the calibration did not complete successfully. <NR1> = 0 indicates that the calibration completed without errors.
<b>Examples</b>	*CAL? starts the internal signal path calibration and might return 0 to indicate that the calibration was successful.

## CALibrate:FACTory (No Query Form)

This command provides the controls for starting and stopping the factory calibration process. The factory calibration process consists of a series of steps.

<b>Group</b>	Calibration and Diagnostic
<b>Syntax</b>	CALibrate:FACTory {START CONTinue PREVIOUS ABOrt DUMp}
<b>Arguments</b>	<p>START initializes the factory calibration sequence and starts the first calibration step.</p> <p>CONTinue begins the next factory calibration step.</p> <p>PREVIOUS tries to run the most recent factory calibration step again.</p> <p>ABOrt stops the calibration process.</p>

DUMP stops the calibration process and prints the calibration constants.

**Examples**     `CALIBRATE:FACTORY START` starts the calibration process.

## **CALibrate:FACTory:NOTIfy:DUE? (Query Only)**

Queries if calibration is due.

**Group**     Calibration and Diagnostic

**Syntax**     `CALibrate:FACTory:NOTIfy:DUE?`

**Returns**     `<NR1> = 1` indicates the oscilloscope is due for calibration.  
`<NR1> = 0` indicates the oscilloscope is not due for calibration.

**Examples**     `CALIBRATE:FACTORY:NOTIFY:DUE?` might return 0 indicating that factory calibration is not due.

## **CALibrate:FACTory:NOTIfy:HOURS**

Sets or returns hours when calibration is due.

**Group**     Calibration and Diagnostic

**Syntax**     `CALibrate:FACTory:NOTIfy:HOURS {INFINite|<NR1>}`  
`CALibrate:FACTory:NOTIfy:HOURS?`

**Arguments**     `<NR1>` is the number of hours, 500 to 10000, before the instrument displays a notifier saying that calibration is due. Numbers entered are rounded to the nearest valid number.

`INFINite` disables the calibration due notifier display feature.

**Examples**     `CALIBRATE:FACTORY:NOTIFY:HOURS 1500` sets the calibration notification time to 1500 hours.

`CALIBRATE:FACTORY:NOTIFY:HOURS?` might return `:CALIBRATE:FACTORY:NOTIFY:HOURS 2000` indicating that the calibration due notifier will be displayed 2000 hours after the last calibration.

## CALibrate:FAcTory:NOTIfy:YEARS

Sets or returns the number of years when calibration is due.

**Group** Calibration and Diagnostic

**Syntax** CALibrate:FAcTory:NOTIfy:YEARS {INFInite|<NR3>}  
CALibrate:FAcTory:NOTIfy:YEARS?

**Arguments** <NR3> is the number of years, 0.5 to 5, before the instrument displays a notifier saying that calibration is due. Numbers entered are rounded to the nearest valid number.

INFInite disables the calibration due notifier display feature.

**Examples** CALIBRATE:FACTORY:NOTIFY:YEARS 1.5e0 sets the calibration due time to 1.5 years.

CALIBRATE:FACTORY:NOTIFY:YEARS? might return CALIBRATE:FACTORY:NOTIFY:YEARS 1.5E0 indicating that the calibration due notifier will be displayed 1.5 years after the last calibration.

## CALibrate:FAcTory:STATus? (Query Only)

Returns the factory calibration status value saved in nonvolatile memory.

**Group** Calibration and Diagnostic

**Syntax** CALibrate:FAcTory:STATus?

**Examples** CALIBRATE:FACTORY:STATUS? might return CALIBRATE:FACTORY:STATUS PASS indicating that factory calibration passed.

## CALibrate:FAcTory:STEPSTatus? (Query Only)

Returns pass during oscilloscope power on without regard to actual calibration status. Returns pass or fail for the factory calibration step that most recently executed when factory calibration is in progress. Use this query to synchronize programmed factory calibration steps.

**Group** Calibration and Diagnostic

**Syntax**     `CALibrate:FACTory:STEPStatus?`

**Examples**     `CALIBRATE:FACTORY:STEPSTATUS?` might return  
`CALIBRATE:FACTORY:STEPSTATUS FAIL` if the last  
oscilloscope calibration step did not pass.

## **CALibrate:FACTory:STEPSTIMulus? (Query Only)**

Returns information about an input signal required for calibration.

**Group**     Calibration and Diagnostic

**Syntax**     `CALibrate:FACTory:STEPSTIMulus?`

**Examples**     `CALIBRATE:FACTORY:STEPSTIMULUS?` might return “2 volts on channel  
1”, indicating a 2 V signal should be applied to channel 1.

## **CALibrate:INTERNAL (No Query Form)**

This command (no query form) starts the internal signal path calibration (SPC) of the instrument. You can use the [CALibrate:INTERNAL:STATUS?](#) query to return the current status of the internal signal path calibration of the instrument.

**Group**     Calibration and Diagnostic

**Syntax**     `CALibrate:INTERNAL`

**Related Commands**     [CALibrate:RESults:SPC?](#)

**Examples**     `CALIBRATE:INTERNAL` starts the internal signal path calibration of the  
instrument.

## CALibrate:INTERNAL:START (No Query Form)

This command (no query form) starts the internal signal path calibration (SPC) of the instrument. This command is the same as the [CALibrate:INTERNAL](#) command. You can use the [CALibrate:INTERNAL:STATUS?](#) query to return the current status of the internal signal path calibration of the instrument.

**Group** Calibration and Diagnostic

**Syntax** CALibrate:INTERNAL:START

**Related Commands** [CALibrate:RESults:SPC?](#)

**Examples** CALIBRATE:INTERNAL:START starts the internal signal path calibration of the instrument.

## CALibrate:INTERNAL:STATUS? (Query Only)

This query-only command returns the current status of the instrument internal signal path calibration for the last SPC operation.

**Group** Calibration and Diagnostic

**Syntax** CALibrate:INTERNAL:STATUS?

**Related Commands** [CAL?](#)

**Returns**

This query will return one of the following:

- INIT indicates the instrument has not had internal signal path calibration run.
- WARMUP indicates the instrument has not warmed up for the required time to perform signal path calibration.
- PASS indicates that the signal path calibration completed successfully.
- FAIL indicates that the signal path calibration did not complete successfully.
- TEMPDRIFT indicates that the instrument has not reached a state of stable temperature adequate for performing signal path calibration.
- RUNNING indicates that the signal path calibration is currently running.

**Examples**

CALIBRATE:INTERNAL:STATUS? might return  
:CALIBRATE:INTERNAL:STATUS INIT indicating that the  
current status of the internal signal path calibration is that it has not been run.

## CALibrate:RESults? (Query Only)

This query-only command returns the status of internal and factory calibrations, without performing any calibration operations. The results returned do not include the calibration status of attached probes. The CALibrate:RESults? query is intended to support GO/NoGO testing of the oscilloscope calibration readiness: all returned results should indicate PASS status if the instrument is "fit for duty". It is quite common, however, to use uncalibrated probes (particularly when the instrumentt inputs are connected into a test system with coaxial cables).

This command is equivalent to selecting Instrument Calibration from the Utilities menu and then viewing the contents of the Status field.

**Group** Calibration and Diagnostic

**Syntax** CALibrate:RESults?

**Related Commands** [CAL?](#)

**Examples**

CALIBRATE:RESULTS? returns the status of all the calibration subsystems. The query might return :CALIBRATE:RESULTS:SPC PASS;FACTORY PASS;FACTORY:VOLTAGE PASS;FREQUENCY PASS;TRIGGER PASS

## CALibrate:RESults:FACTory? (Query Only)

This query-only command returns the status of internal and factory calibration, without performing any calibration operations.

**Group** Calibration and Diagnostic

**Syntax** CALibrate:RESults:FACTory?

**Examples** CALIBRATE:RESULTS:FACTORY? might return :CALIBRATE:RESULTS:FACTORY FREQUENCY PASS; TRIGGER PASS; VOLTAGE PASS.

## CALibrate:RESults:FACTory:Frequency? (Query Only)

This query-only command returns the status of internal and factory frequency calibration, without performing any calibration operations.

**Group** Calibration and Diagnostic

**Syntax** CALibrate:RESults:FACTory:Frequency?

**Examples** CALIBRATE:RESULTS:FACTORY:FREQUENCY? might return :CALIBRATE:RESULTS:FACTORY:FREQUENCY PASS.

## CALibrate:RESults:FACTory:Trigger? (Query Only)

This query-only command returns the status of internal and factory trigger calibration, without performing any calibration operations.

**Group** Calibration and Diagnostic

**Syntax** CALibrate:RESults:FACTory:Trigger?

**Examples** CALIBRATE:RESULTS:FACTORY:TRIGGER? might return :CALIBRATE:RESULTS:FACTORY:TRIGGER PASS.



## CALibrate:RESuLts:FACTory:Voltage? (Query Only)

This query-only command returns the status of internal and factory voltage calibration, without performing any calibration operations.

**Group** Calibration and Diagnostic

**Syntax** CALibrate:RESuLts:FACTory:Voltage?

**Examples** CALIBRATE:RESULTS:FACTORY:VOLTAGE? might return  
:CALIBRATE:RESULTS:FACTORY:VOLTAGE PASS.

## CALibrate:RESuLts:SPC? (Query Only)

This query-only command returns the status of the SPC operation. However, this query does not cause an SPC to be run.

**Group** Calibration and Diagnostic

**Syntax** CALibrate:RESuLts:SPC?

**Related Commands** [CAL?](#)

**Returns** INIT indicates that  
PASS indicates that the last SPC operation passed.  
FAIL indicates that the last SPC operation failed.  
SEQERROR  
RUNNING indicates that the SPC operation is running.

**Examples** CALIBRATE:RESULTS:SPC? returns the results of the last SPC operation: either PASS or FAIL.

## CALibrate:TEMPerature? (Query Only)

Returns the temperature during the last signal path compensation operation.

**Group** Calibration and Diagnostic

**Syntax**     CALibrate:TEMPerature?

**Examples**     CALIBRATE:TEMPERATURE? might return :CALIBRATE:TEMPERATURE  
2.3E01 indicating that the instrument temperature during the last SPC operation  
was 23 °C.

## CH<x>? (Query Only)

This query-only command returns the vertical parameters for the specified channel. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is similar to selecting Vertical Setup from the Vertical menu.

**Group**     Vertical

**Syntax**     CH<x>?

**Examples**     CH1? might return the following vertical parameters for  
channel 1: :CH1:BANDWIDTH 1.0000E+09;COUPLING  
DC;DESKEW0.0000E+00;OFFSET 0.0000E+00;POSITION  
0.0000E+00;SCALE 5.0000E-01;TERMINATION 1.0000E+06;PROBCAL  
INIT;PROBE:GAIN 1.0000E+00;RESISTANCE 1.0000E+06;UNITS  
"V";ID:TYPE "1X";SERNUMBER "N/A";CH1:PROBEFUNC:EXTATTEN  
1.0000E+00;EXTUNITS "None";CH1:LABEL:NAME "";XPOS 5;YPOS 5

## CH<x>:BANDwidth

This command sets or returns the selectable low-pass bandwidth limit filter of the specified channel. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This is equivalent to selecting Bandwidth from the Vertical menu.

The query form of this command always returns the maximum bandwidth of the channel.

**Group**     Vertical

**Syntax**     CH<x>:BANDwidth {TWEnty|TWOfifty|FUL1|<NR3>}  
CH<x>:BANDwidth?

**Arguments**     **TwEnty** sets the upper bandwidth limit to 20 MHz.

**TwoFifty** sets the upper bandwidth limit to 250 MHz.

**FUL1** disables any optional bandwidth limiting. The specified channel operates at its maximum attainable bandwidth.

**<NR3>** argument is a double-precision ASCII string. The instrument rounds this value to an available bandwidth using geometric rounding and then uses this value set the upper bandwidth.

**Examples**     **CH1:BANDWIDTH TWENTY** sets the bandwidth of channel 1 to 20 MHz.

**CH2:BANDWIDTH?**, for TDS5052B and TDS5054B instruments, might return **:CH2:BANDWIDTH 500.0000E+06** indicating that there is no bandwidth limiting on channel 2.

For TDS5104 instruments, this query might return **:CH2:BANDWIDTH 1.0000E+09** indicating that there is no bandwidth limiting on channel 2.

## CH<x>:COUPLing

This command sets or returns the input attenuator coupling setting for the specified channel. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is equivalent to selecting Coupling from the Vertical menu.

**Group**     Vertical

**Syntax**     **CH<x>:COUPLing {AC|DC|GND}**  
**CH<x>:COUPLing?**

**Arguments**     **AC** sets the specified channel to AC coupling.

**DC** sets the specified channel to DC coupling.

**GND** sets the specified channel to ground. Only a flat, ground-level waveform will be displayed.

**Examples**     **CH2:COUPLING GND** sets channel 2 to ground.

**CH3:COUPLing?** might return **:CH3:COUPLing DC** indicating that channel 3 is set to DC coupling.

## CH<x>:DESKew

This command sets or returns the deskew time for the specified channel. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is equivalent to selecting Deskew from the Vertical Setup menu.

You can adjust the deskew time to add an independent, channel-based delay time to the delay (set by the horizontal position control and common to all channels) from the common trigger point to first sample taken for each channel. This lets you compensate individual channels for different delays introduced by their individual input hook ups.

**Group** Vertical

**Syntax** CH<x>:DESKew <NR3>  
CH<x>:DESKew?

**Arguments** <NR3> is the deskew time for this channel, ranging from -25 ns to +25 ns with a resolution of 1 ps. Out-of-range values are clipped.

**Examples** CH4:DESKew 5.0E-9 sets the deskew time for channel 4 to 5 ns.  
CH2:DESKew? might return :CH2:DESKew 2.0000E-09 indicating that the deskew time for channel 2 is set to 2 ns.

## CH<x>:INVert

This command sets or returns the invert function for the specified channel. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. When on, the invert function inverts the waveform for the specified channel. This command is equivalent to selecting On or Off for the Invert function in the Vertical Setup control window.

---

**NOTE.** This command inverts the waveform for display purposes only. The instrument does not use an inverted waveform for triggers or trigger logic inputs.

---

**Group** Vertical

**Syntax** CH<x>:INVert {ON|OFF|NR1}  
CH<x>:INVert?

**Arguments** OFF argument turns off the invert function for the specified channel.

ON argument turns on the invert function for the specified channel.

<NR1> = 0 turns off the invert function; any other value turns on the invert function.

**Examples** CH4:INVERT ON inverts the waveform on channel 4.

CH2:INVERT? might return :CH2:INVERT 0 indicating that the invert function for channel 2 is off.

## CH<x>:OFFSet

This command sets or returns the vertical offset for the specified channel. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is equivalent to selecting Offset from the Vertical menu.

This command offsets the vertical acquisition window (moves the level at the vertical center of the acquisition window) for the selected channel. Visualize offset as scrolling the acquisition window towards the top of a large signal for increased offset values, and scrolling towards the bottom for decreased offset values. The resolution of the vertical window sets the offset increment for this control.

Offset adjusts only the vertical center of the acquisition window for channel waveforms to help determine what data is acquired. The instrument always displays the input signal minus the offset value. The channel reference marker will move to the vertical graticule position given by the negative of the offset value divided by the scale factor, unless that position is off-screen. If the computed coordinate for the reference mark is off-screen, the mark moves to the nearest screen limit and changes from a right-pointing arrow (→) to an arrow pointing in the appropriate off-screen direction.

The channel offset range is either  $\pm 100$  V,  $\pm 10$  V or  $\pm 1.0$  V, depending on the vertical scale factor.

**Table 2-31: Vertical Scale Adjust Factors**

	From 0.001 to 0.1	From 0.101 to 1.0	From 1.01 to 100
When internal scale adjust gain is:			
Product of Offset range, Probe gain, and transducer gain is:	$\pm 1.0$ V	$\pm 10.0$ V	$\pm 100.0$ V

**NOTE.** The above table describes instrument behavior only when no probe is attached, and when the external attenuation factor is 1.0.

<b>Group</b>	Vertical
<b>Syntax</b>	CH<x>:OFFSet <NR3> CH<x>:OFFSet?
<b>Related Commands</b>	<a href="#">CH&lt;x&gt;:POSition</a>
<b>Arguments</b>	<NR3> is the offset value for the specified channel.
<b>Examples</b>	CH3:OFFSet 2.0E-3 sets the offset for channel 3 to 2 mV.  CH4:OFFSet? might return :CH4:OFFSET 1.0000E-03 indicating that the offset for channel 4 is set to 1 mV.

## CH<x>:POSition

This command sets or returns the vertical position of the specified channel. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. The position value is applied to the signal before it is digitized. This command is equivalent to selecting Position/Scale from the Vertical menu and either viewing or setting Position.

Increasing the position value of a waveform causes the waveform to move up, and decreasing the position value causes the waveform to move down. Position adjusts only the display position of a waveform, whether it is a channel, math, or reference waveform. The position value determines the vertical graticule coordinate at which input signal values, minus the present offset setting for that channel, are displayed. For example, if the position for Channel 3 is set to 2.0 and the offset is set to 3.0, then input signals equal to 3.0 units are displayed 2.0 divisions above the center of the screen.

<b>Group</b>	Vertical
<b>Syntax</b>	CH<x>:POSition <NR3> CH<x>:POSition?
<b>Related Commands</b>	<a href="#">CH&lt;x&gt;:OFFSet</a> , <a href="#">REF&lt;x&gt;:VERTical:POSition</a> , <a href="#">MATH[1]:VERTical:POSition</a>
<b>Arguments</b>	<NR3> is the position value, in divisions from the center graticule, ranging from 8 to -8 divisions.

**Examples** CH2:POSition 1.3E+00 positions the Channel 2 input signal 1.3 divisions above the center graticule.

CH1:POSition? might return :CH1:POSITION -1.3000E+00 indicating that the current position of Channel 1 is 1.3 divisions below the center graticule.

## CH<x>:PRObe? (Query Only)

This query-only command returns all information concerning the probe that is attached to the specified channel. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments.

**Group** Vertical

**Syntax** CH<x>:PRObe?

**Examples** CH2:PROBE? might return :CH2:PROBE:GAIN 1.0000E-01; RESISTANCE 1.0000E+07; UNITS "V"; ID:TYPE "10X" 'SERNUMBER "N/A" for a 10x probe, indicating that (among other parameters) the attenuation factor for the probe attached to channel 2 is 100.0 mV (assuming that probe units are set to volts).

## CH<x>:PRObe:AUTOZero (No Query Form)

This command executes the probe auto-zero operation.

**Group** Vertical

**Syntax** CH<x>:PRObe:AUTOZero EXECute

## CH<x>:PRObe:DEGAUss (No Query Form)

This command executes a probe degauss operation.

**Group** Vertical

**Syntax** CH<x>:PRObe:DEGAUSS EXECute

## CH<x>:PRObe:DEGAUss:STATE? (Query Only)

This query-only command returns the probe degauss state.

**Group** Vertical

**Syntax** CH<x>:PRObe:DEGAUss:STATE?

## CH<x>:PRObe:FORCERange

This command sets

**Group** Vertical

**Syntax** CH<x>:PRObe:FORCERange <NR3>  
CH<x>:PRObe:FORCERange?

## CH<x>:PRObe:GAIN

This command sets or returns the gain factor of the probe that is attached to the specified channel. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. The "gain" of a probe is the output divided by the input transfer ratio. For example, a common 10x probe has a gain of 0.1.

**Group** Vertical

**Syntax** CH<x>:PRObe:GAIN <NR3>  
CH<x>:PRObe:GAIN?

**Related Commands** [CH<x>:SCAle](#)

**Arguments** <NR3> is the probe gain. Allowed values depend on the specific probe.

**Examples** CH2:PROBE:GAIN? might return :CH2:PROBE:GAIN 0.1000E+00 indicating that the attached 10x probe delivers 0.1 V to the channel 2 BNC for every 10 V applied to the probe input.



## CH<x>:PRObe:ID? (Query Only)

This query-only command returns the type and serial of the probe that is attached to the specified channel. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments.

**Group** Vertical

**Syntax** CH<x> : PRObe : ID?

**Examples** CH2:PROBE:ID? might return :CH2:PROBE:ID:TYPE "10X";SERNUMBER "N/A" indicating that a passive 10x probe of unknown serial number is attached to channel 2.

## CH<x>:PRObe:ID:SERNuMBER? (Query Only)

This query-only command returns the serial number of the probe that is attached to the specified channel. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments.

---

**NOTE.** For Level 0 and 1 probes, the serial number will be "N/A".

---

**Group** Vertical

**Syntax** CH<x> : PRObe : ID : SERNuMBER?

**Examples** CH1:PROBE:ID:SERNUMBER? might return :CH1:PROBE:ID:SERNUMBER "B010289" indicating that the serial number of the probe attached to channel 1 is B010289.

## CH<x>:PRObe:ID:TYPE? (Query Only)

This query-only command returns the type of probe that is attached to the specified channel. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. Level 2 (or higher) probes supply their exact product nomenclature; for Level 0 or 1 probes, a generic 'type string' of "nnX" is returned.

**Group** Vertical

**Syntax** CH<x>:PRObe:ID:TYPE?

**Examples** CH1:PROBE:ID:TYPE? might return :CH1:PROBE:ID:TYPE "P6203" indicating that P6203-type probe is attached to channel 1.

## CH<x>:PRObe:RESistance? (Query Only)

This query-only command returns the resistance factor of the probe that is attached to the specified channel. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments.

**Group** Vertical

**Syntax** CH<x>:PRObe:RESistance?

**Examples** CH2:PROBE:RESISTANCE? might return :CH2:PROBE:RESISTANCE 10.0000E+06 indicating that the input resistance of the probe attached to Channel 2 is 10 MΩ.

## CH<x>:PRObe:SIGnal

**Group** Vertical

**Syntax** CH<x>:PRObe:SIGna1 {BYPass|PASS}  
CH<x>:PRObe:SIGna1?

## CH<x>:PRObe:UNIts? (Query Only)

This query-only command returns a string describing the units of measure for the probe attached to the specified channel. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments.

**Group** Vertical

**Syntax** CH<x>:PRObe:UNITs?

**Examples** CH4:PROBE:UNITS? might return :CH4:PROBE:UNITS "V" indicating that the units of measure for the probe attached to channel 4 are volts.

## CH<x>:SCAlE

This command sets or returns the vertical scale of the specified channel. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. Sending this command is equivalent to selecting Vertical Setup from the Vertical menu and then viewing or setting the Scale.

Each waveform has a vertical scale parameter. For a signal with constant amplitude, increasing the Scale causes the waveform to be displayed smaller. Decreasing the scale causes the waveform to be displayed larger.

Scale affects all waveforms, but affects channel waveforms differently from other waveforms:

- For channel waveforms, this setting controls the vertical size of the acquisition window as well as the display scale. The range and resolution of scale values depends on the probe attached and any other external factors you have specified.
- For reference and math waveforms, this setting controls the display only, graphically scaling these waveforms and having no effect on the acquisition hardware.

**Group** Vertical

**Syntax** CH<x>:SCAlE <NR3>  
CH<x>:SCAlE?

**Related Commands** CH<x>:OFFSet, CH<x>:POSition, REF<x>:VERTical:SCAlE, MATH[1]:VERTical:SCAlE

**Arguments** <NR3> is the vertical channel scale in units per division.

**Examples** CH4:SCALE 100E-03 sets the channel 4 scale to 100 mV per division.

CH2:SCALE? might return :CH2:SCALE 1.0000E+00 indicating that the current scale setting of channel 2 is 1 V per division.

## CH<x>:TERmination

This command sets the connected/disconnected status of a 50  $\Omega$  resistor, which may be connected between the specified channel's coupled input and instrument ground. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. There is also a corresponding query that requests the termination parameter and translates this enumeration into one of the two float values. This command is equivalent to selecting Termination from the Vertical menu or toggling between termination values from the VERTICAL area of the front panel.

**Group** Vertical

**Syntax** CH<x>:TERmination <FIFTy|MEG|NR3>  
CH<x>:TERmination?

**Arguments** FIFTy sets the channel input resistance to 50  $\Omega$ .  
MEG sets the channel input resistance to 1 M $\Omega$ .  
<NR3> specifies the channel input resistance, which can be specified as 50 or 1,000,000  $\Omega$ .

**Examples** CH4:TERMINATION 50.0E+0 establishes 50  $\Omega$  impedance on channel 1.  
CH2:TERMINATION? might return :CH2:TERMINATION 50.0E+0 indicating that channel 2 is set to 50  $\Omega$  impedance.

## CH<x>:YUNits

This command sets or returns the units for the specified channel to a custom string. String arguments are case insensitive and any unsupported units will generate an error.

**Group** Vertical

**Syntax** CH<x>:YUNits <QString>  
CH<x>:YUNits?

## CLEARMenu (No Query Form)

Clears the current menu from the display. This command is equivalent to pressing the front panel MENU OFF button.

**Group** Miscellaneous

**Syntax** CLEARMenu

## **\*CLS (No Query Form)**

This command (no query form) clears the following:

- Event Queue
- Standard Event Status Register
- Status Byte Register (except the MAV bit)

If the \*CLS command immediately follows an <EOI>, the Output Queue and MAV bit (Status Byte Register bit 4) are also cleared. MAV indicates that information is in the output queue. The device clear (DCL) GPIB control message will clear the output queue and thus MAV. \*CLS does not clear the output queue or MAV.

\*CLS can suppress a Service Request that is to be generated by an \*OPC. This will happen if a single sequence acquisition operation is still being processed when the \*CLS command is executed.

**Group** Status and Error

**Syntax** \*CLS

**Related Commands** [DESE](#), [\\*ESE](#), [\\*ESR?](#), [EVENT?](#), [EVMsg?](#), [\\*SRE](#), [\\*STB?](#)

**Examples** \*CLS clears the instrument status data structures.

## **CURSor?**

This query-only command returns all of the current cursor settings.

**Group** Cursor

**Syntax** CURSor?

**Examples**      `CURSOR?` might return the following as the current cursor settings:  
`:CURSOR:STATE 1;FUNCTION VBARS;LINESTYLE SOLID;MODE  
INDEPENDENT;SOURCE1 CH1; SOURCE2 CH1;SCREEN:STYLE  
LINES; :CURSOR:VBARS:UNITS SECONDS;POSITION1  
-1.6000E-6;POSITION2 1.6000E-6; :CURSOR:HBARS:POSITION1  
300.0000E-3; POSITION2 -300.0000E-3;UNITS BASE;  
:CURSOR:WAVEFORM:POSITION1 -1.6000E-6;POSITION2 1.6000E-6;  
SOURCE2 CH1;UNITS BASE;STYLE LINE_X;:CURSOR:XY :PRODDELTA  
99.1000E+36;PRODUCT1 99.1000E+36;PRODUCT2 99.1000E+36;  
READOUT RECTANGULAR;RADIUS1 99.1000E+36;RADIUS2  
99.1000E+36; RATDELTA 99.1000E+36;RATIO1 99.1000E+36;RATIO2  
99.1000E+36; RDELTA 99.1000E+36;RECTX1 0.0000;RECTX2  
0.0000;RECTY1 0.0000; RECTY2 0.0000;THDELTA 99.1000E+36;THETA1  
99.1000E+36; THETA2 99.1000E+36;XDELTA 0.0000;YDELTA 0.0000`

## CURSor:FUNctioN

This command sets or returns the cursor type. Sending this command is equivalent to selecting Cursor Type from the Cursors menu, and then choosing from the drop-down list.

**Group**      Cursor

**Syntax**      `CURSor:FUNctioN {OFF|SCREEN|WAVEform}`  
`CURSor:FUNctioN?`

**Arguments**      `OFF` removes the cursors from the display but does not change the cursor type.

`SCREEN` specifies both horizontal and vertical bar cursors, which measure in horizontal and vertical units specified by the Cursor 1 and Cursor 2 Sources. Use these cursors to measure anywhere in the waveform display area.

`WAVEform` specifies paired or split cursors in YT display format for measuring waveform amplitude and time. In XY and XYZ format, these cursors indicate the amplitude positions of an XY pair (Ch1 vs Ch2 voltage, where Ch1 is the X axis and Ch2 is the Y axis) relative to the trigger.

**Examples**      `CURSOR:FUNCTION VBARS` selects the vertical bar cursors type

`CURSOR:FUNCTION?` might return `:CURSor:FUNctioN SCREEN` indicating that the screen cursors are currently selected.

## CURSor:HBArS? (Query Only)

This query-only command returns the current settings for the horizontal bar cursors.

**Group** Cursor

**Syntax** CURSor:HBArS?

**Examples** CURSOR:HBARS? might return the horizontal bar setting as :CURSOR:HBARS:POSITION1 320.0000E-03;POSITION2-320.0000E-03;UNITS BASE

## CURSor:HBArS:DELTA? (Query Only)

This query-only command returns the vertical difference between the two horizontal bar cursors.

**Group** Cursor

**Syntax** CURSor:HBArS:DELTA?

**Related Commands** [CURSor:HBArS:UNItS](#)

**Returns** A floating point value with an exponent.

**Examples** CURSOR:HBARS:DELTA? might return :CURSOR:HBARS:DELTA 5.0800E+00 indicating that the voltage difference between the two cursors is 5.08 V.

## CURSor:HBArS:POSITION<x>

This command sets or returns the horizontal bar cursor position relative to ground, which is expressed in vertical units (usually volts). The cursor is specified by x, which can be 1 or 2. This command is the equivalent to selecting Cursor Position from the Cursors menu, selecting the H Bars Cursor Type, and then viewing or editing the desired cursor position.

**Group** Cursor

**Syntax**      `CURSor:HBARS:POSITION<x> <NR3>`  
`CURSor:HBARS:POSITION<x>?`

**Related Commands**      [CURSor:FUNCtion](#)

**Arguments**      <NR3> specifies the cursor position relative to ground.

**Examples**      `CURSOR:HBARS:POSITION1 25.0E-3` positions Cursor 1 of the horizontal cursors at 25 mV.  
`CURSOR:HBARS:POSITION2?` might return `:CURSOR:HBARS:POSITION2 -64.0000E-03` indicating that Cursor 2 of the horizontal bar cursors is at -64 mV.

## CURSor:HBARS:UNIts

This command sets or returns the units for the horizontal bar cursors.

**Group**      Cursor

**Syntax**      `CURSor:HBARS:UNIts {BASE|PERcent}`  
`CURSor:HBARS:UNIts?`

**Arguments**      BASE selects the units current selected by the Vbar cursors.  
 PERcent selects ratio cursors.

**Examples**      `CURSOR:HBARS:UNITS` might return `:CURSOR:HBARS:UNITS BASE` indicating that the units for the horizontal bar cursors are base.

## CURSor:HBARS:USE (No Query Form)

Sets the horizontal bar cursor measurement scale. This command is only applicable when ratio cursors are on.

**Group**      Cursor

**Syntax**      `CURSor:HBARS:USE {CURrent|FIVEdivs}`

**Related Commands**      [CURSor:HBARS:UNIts](#)



- Arguments**     **CURrent** sets the H Bar measurement scale so that 0% is the current position of the lowest H Bar cursor and 100% is the current position of the highest H Bar cursor. Moving the cursors after executing this command does not change the 0% or 100% scale points.
- FIVEdivs** sets H Bar measurement scale so that 5 screen major divisions is 100%, where 0% is -2.5 divisions and 100% is +2.5 divisions from the center horizontal graticule.
- Examples**     **CURSOR:HBars:USE FIVEDIVS** sets the H Bar measurement scale so that 5 screen major divisions equals 100%.

## **CURSor:MODe**

This command sets or returns whether the two cursors move together in unison or separately. This command is the equivalent of selecting Cursor Mode from the Cursors menu, and then choosing from the drop-down list.

- Group**     Cursor
- Syntax**     **CURSor:MODe** {**TRACK**|**INDEpendent**}  
                  **CURSor:MODe?**
- Arguments**     **TRACK** ties the navigational functionality of the two cursors together. For cursor 1 adjustments, this ties the movement of the two cursors together; however, cursor 2 continues to move independently of cursor 1.
- INDEpendent** allows independent adjustment of the two cursors.
- Examples**     **CURSOR:MODE TRACK** specifies that the cursor positions move in unison.
- CURSOR:MODE?** might return **:CURSOR:MODE TRACK** indicating that the two cursors move in unison.

## **CURSor:VBArS? (Query Only)**

This command returns the current settings for the vertical bar cursors.

- Group**     Cursor
- Syntax**     **CURSor:VBArS??**

**Related Commands**     [DATa:START](#), [DATa:STOP](#)

**Examples**     `CURSOR:VBARS??` might return the following vertical bar settings  
                   `:CURSOR:VBARS:UNITS SECONDS;POSITION1 1.0000E-06;POSITION2`  
                   `9.0000E-06`

## **CURSor:VBArS:DELTA? (Query Only)**

This query-only command returns the horizontal difference between the two vertical bar cursors. The units are specified by the `CURSor:VBArS:UNItS` command. The position can appear in units of Base or 1/Base.

**Group**     Cursor

**Syntax**     `CURSor:VBArS:DELTA?`

**Related Commands**     [CURSor:VBArS:UNItS](#)

**Returns**     A floating point value with an exponent.

**Examples**     `CURSOR:VBARS:DELTA?` might return `:CURSOR:VBARS:DELTA 1.0640E+00` indicating that the time between the vertical bar cursors is 1.064 s.

## **CURSor:VBArS:HPOS<x>? (Query Only)**

Returns the horizontal value of the specified vertical bar ticks.

**Group**     Cursor

**Syntax**     `CURSor:VBArS:HPOS<x>?`

**Related Commands**     [CURSor:VBArS:UNItS](#)

**Arguments**     `<x>` specifies the cursor. Valid values are 1 and 2.

**Returns**     `<NR3>` indicates the value of one of the tics. The units are specified by the `CURSor:VBArS:UNItS` command.

**Examples**     `CURSor:VBArS:HPOS2?` might return `CURSOR:VBARS:HPOS2 1.00E-6`, indicating the value of one vertical bar tick.

## **CURSor:VBArS:POSITION<x>**

This command sets or returns the horizontal position for vertical bar cursors. The cursor is specified by *x*, which can be 1 or 2. Values are with respect to trigger position or the zero reference point for the designated waveform (if horizontal units are not set to time). Use the `CURSor:VBArS:UNIts` command to specify units. The position can appear in units of base or 1/base.

This command is the equivalent of selecting Cursor Setup from the Cursors menu, selecting V Bars Cursor Type, and then viewing or editing the desired cursor position.

**Group**     Cursor

**Syntax**     `CURSor:VBArS:POSITION<x> <NR3>`  
`CURSor:VBArS:POSITION<x>?`

**Related Commands**     [CURSor:VBArS:UNIts](#)

**Arguments**     *<NR3>* specifies the cursor position.

**Returns**     A floating point value with an exponent.

**Examples**     `CURSor:VBArS:POSITION2 9.00E-6` positions the cursor2 vertical bar cursor at 9 ms.

`CURSor:VBArS:POSITION1?` this command might return  
`:CURSOR:VBARS:POSITION1 1.0000E-06` indicating that the  
cursor1 vertical bar is positioned at 1 ms.

## **CURSor:VBArS:UNIts**

This command sets or returns the units for the vertical bar cursors.

**Group**     Cursor

**Syntax**     `CURSor:VBARs:UNITs {SECOnds|HERtz|DEGrees|PERcent}`  
`CURSor:VBARs:UNITs?`

**Arguments**     `SECOnds` sets the units of the vertical bar cursors for the time domain (seconds).  
`HERtz` sets the units of the vertical bar cursors for the frequency domain (Hertz).  
`DEGrees` sets the units to degrees for use with an XY display.  
`PERcent` sets the units to percent for use with ratio cursors.

**Returns**     A string for `SECOnds` or `HERtz`, depending on the current vertical bar cursor units.

**Examples**     `CURSOR:VBARS:UNITS HERTz` sets the units of the VBARS cursors to 1/seconds.  
`CURSOR:VBARS:UNITS?` might return `:CURSOR:VBARS:UNITS SECONDS` indicating that the units for the vertical bar cursor are currently set to seconds.

## **CURSor:VBARs:USE (No Query Form)**

Sets the vertical bar cursor measurement scale. This command only applicable when ratio cursors are on.

**Group**     Cursor

**Syntax**     `CURSor:VBARs:USE {CURrent|FIVEDivs}`

**Related Commands**     [CURSor:VBARs:UNITs](#)

**Arguments**     `CURrent` sets the V Bar measurement scale so that 0% is the current position of the left-most V Bar cursor and 100% is the current position of the right-most V Bar cursor. Moving the cursors after executing this command does not change the 0% or 100% scale points.  
`FIVEDivs` sets V Bar measurement scale so that 5 screen major divisions is 100%, where 0% is -2.5 divisions and 100% is +2.5 divisions from the center vertical graticule.

**Examples**     `CURSOR:VBARS:USE CURRENT` sets the V Bar measurement scale to use the current cursor positions as 0% and 100% of scale if units are set to %.

## CURSor:VBArS:VDELtA? (Query Only)

Returns the vertical difference between the two vertical bar cursor ticks. The units are specified by the CURSor:HBArS:UNItS query.

**Group** Cursor

**Syntax** CURSor:VBArS:VDELtA?

**Related Commands** [CURSor:HBArS:UNItS](#)

**Returns** <NR3> indicates the horizontal difference between the two vertical bar cursors.

**Examples** CURSOR:VBARS:VDELTA? might return CURSOR:VBARS:VDELTA 1.064E+0, indicating that the vertical difference between the vertical bar cursors ticks is 1.064 units.

## CURVe

This command transfers waveform data to and from the instrument. Each waveform that is transferred has an associated waveform preamble that contains information such as data format and scale.

The CURVe? query transfers data from the instrument. The data source is specified by the [DATA:SOURce](#) command. The first and last data points are specified by the [DATA:START](#) and [DATA:STOP](#) commands.

The CURVe command transfers waveform data to the instrument. The data is stored in the reference memory location specified by [DATA:DESTination](#), starting with the data point specified by [DATA:START](#). Only one waveform can be transferred at a time. The waveform will only be displayed if the reference is displayed.

**Group** Waveform Transfer

**Syntax** CURVe {<Block>|<asc curve>}  
CURVe?

**Related Commands**     [DATA:DESTination](#), [DATA:SOUrce](#), [DATA:STARt](#), [DATA:STOP](#),  
[SAVe:WAVEform](#), [SAVe:WAVEform:FILEFormat](#), [WFMImpre?](#),  
[WFMImpre:BYT\\_Nr](#), [WFMOuImpre?](#)

**Arguments**     **<Block>** is the waveform data in binary format. The waveform is formatted as:  
 #<x><yyy><data><newline>, where:

<x> is the number of y bytes. For example, if <yyy>=500, then <x>=3)

<yyy> is the number of bytes to transfer. If width is 1, then all bytes on the bus are single data points. If width is 2, then all bytes on the bus are 2-byte pairs. If width is 4, then all bytes on the bus are 4-byte pairs. Use the [WFMImpre:BYT\\_Nr](#) command to set the width.

<data> is the curve data.

<newline> is a single byte new line character at the end of the data.

**<asc curve>** is the waveform data in ASCII format. The format for ASCII data is <NR1>[,<NR1>...], where each <NR1> represents a data point.

**Examples**     CURVE? with ASCII encoding, start and stop of 1 and 10 respectively, and a width set to 1 might return :CURVE 61,62,61,60,60,-59,-59,-58,-58,-59

---

**NOTE.** Curve data is transferred from the instrument asynchronously and, dependent upon the length of the curve record, such transfers can require several seconds to complete. During this time, the instrument will not respond to user controls. You can interrupt these asynchronous data transfers by sending a device clear message to the instrument or by interrupting the query with another command or query. In order to verify that curve data has been completely transferred, it is recommended that you follow such queries with an [\\*ESR?](#) query and verify that the error bit returned and, if set, check the event queue to ascertain the reason for the error. If the error was caused by an interrupted query, then the asynchronous data transfer had not completed when the [\\*ESR?](#) query was sent. In this case, you may need increase your program's time-out value in order to ensure that all data is transferred and read.

---

CURVE **<Block>** sets the format of the waveform data, transferred to and from the instrument, to binary format.

## DATA

This command sets or returns the format and location of the waveform data that is transferred with the [CURVe](#) command.

<b>Group</b>	Waveform Transfer
<b>Syntax</b>	<code>DATA {INIT SNAP}</code> <code>DATA?</code>
<b>Related Commands</b>	<a href="#">CURVe</a> , <a href="#">DATA:START</a> , <a href="#">DATA:STOP</a> , <a href="#">DATA:ENCdg</a> , <a href="#">SAVe:WAVEform:FILEFormat</a> , <a href="#">WFMinpre:NR_Pt</a> , <a href="#">WFMOutpre:NR_Pt?</a>
<b>Arguments</b>	<p>INIT initializes the waveform data parameters to their factory defaults.</p> <p>SNAP sets <code>DATA:START</code> and <code>DATA:STOP</code> to match the current V Bar/Paired/Split cursor positions.</p>
<b>Examples</b>	<p><code>DATA?</code> might return <code>:DATA:DESTINATION REF1:ENCdg RIBINARY;SOURCE CH1;START 1;STOP 500</code></p> <p><code>DATA INIT</code> initializes the waveform data parameters to their factory defaults.</p>

## DATA:DESTination

This command sets or returns the reference memory location for storing waveform data that is transferred into the instrument by the [CURVe](#) command.

<b>Group</b>	Waveform Transfer
<b>Syntax</b>	<code>DATA:DESTination REF&lt;x&gt;</code> <code>DATA:DESTination?</code>
<b>Related Commands</b>	<a href="#">CURVe</a>
<b>Arguments</b>	<p>REF&lt;x&gt; is the reference where the waveform will be stored. The reference number is specified by x, which ranges from 1 through 4.</p>
<b>Examples</b>	<p><code>DATA:DESTINATION?</code> might return <code>:DATA:DESTINATION REF3</code> indicating that reference 3 is the currently selected reference memory location for waveform data. <code>DATA:DESTINATION REF1</code> indicates that incoming waveform data be stored in reference 1.</p>

## DATA:ENCdg

This command sets or returns the format of outgoing waveform data. This command is equivalent to setting [WFMOupre:ENCdg](#), [WFMOupre:BN\\_Fmt](#), and [WFMOupre:BYT\\_Or](#). Setting the DATA:ENCdg value causes the corresponding WFMOupre values to be updated and visa versa.

---

**NOTE.** *Values are constrained (for outbound data) to the format of the data specified by DATA:SOURce.*

---

**Group** Waveform Transfer

**Syntax** DATA:ENCdg  
 {ASCIi|FASTest|RIBinary|RPBinary|UNLocked|SRIBinary  
 |SRPbinary}  
 DATA:ENCdg?

**Related Commands** [WFMOupre:ENCdg](#), [WFMOupre:BN\\_Fmt](#), [WFMOupre:BYT\\_Or](#)



**Arguments**

- **ASCIi** specifies the ASCII representation of signed INT, FLOAT. If ASCII is the value, then :BN\_Fmt and :BYT\_Or are ignored.

- **FASTest** specifies that the data be sent in the fastest possible manner consistent with maintaining accuracy and is interpreted with respect to the first waveform specified in the DATA:SOURce list.

:ENCdg will always be BIN, :BYT\_Or will always be LSB, but :BN\_Fmt and :BYT\_Nr will depend on the first DATA:SOURce waveform. :BN\_Fmt will be RI unless the waveform is internally stored as a floating point number, in which case the FP format will be used.

- **RIBinary** specifies signed integer data point representation with the most significant byte transferred first.

When :BYT\_Nr is 1, the range is from -128 through 127. When :BYT\_Nr is 2, the range is from -32,768 through 32,767. When :BYT\_Nr is 8, then the waveform being queried has been set to Fast Acquisition mode. Center screen is 0 (zero). The upper limit is one division above the top of the screen and the lower limit is one division below the bottom of the screen. This is the default argument.

- **RPBinary** specifies the positive integer data-point representation, with the most significant byte transferred first.

When :BYT\_Nr is 1, the range from 0 through 255. When :BYT\_Nr is 2, the range is from 0 to 65,535. When :BYT\_Nr is 8, then the waveform being queried has been set to Fast Acquisition mode. The center of the screen is 127. The upper limit is one division above the top of the screen and the lower limit is one division below the bottom of the screen.

- **FPBinary** specifies the floating point (width = 4) data.

The range is from  $-3.4 \times 10^{38}$  to  $3.4 \times 10^{38}$ . The center of the screen is 0. The upper limit is one division above the top of the screen and the lower limit is one division below the bottom of the screen.

The **FPBinary** argument is only applicable to math waveforms or ref waveforms saved from math waveforms.

- **SRIBinary** is the same as **RIBinary** except that the byte order is swapped, meaning that the least significant byte is transferred first. This format is useful when transferring data to IBM compatible PCs.
- **SRPBinary** is the same as **RPBinary** except that the byte order is swapped, meaning that the least significant byte is transferred first. This format is useful when transferring data to PCs.
- **SFPBinary** specifies floating point data in IBM PC format. The **SFPBinary** argument only works on math waveforms or ref waveforms saved from math waveforms

**Table 2-32: DATA and WFMOutpre Parameter Settings**

DATA:ENCdg Setting	WFMOutpre Settings			
	:ENCdg	:BN_Fmt	:BYT_Or	:BYT_NR
AScii	ASC	N/A	N/A	1,2,4,8
FAStest	BIN	RI/FP	MSB	1,2,4
RIBinary	BIN	RI	MSB	1,2,8
RPBinary	BIN	RP	MSB	1,2,8
FPBinary	BIN	FP	MSB	4
SRIBinary	BIN	RI	LSB	1,2,8
SRPBinary	BIN	RP	LSB	1,2,8
SFPBinary	BIN	FP	LSB	4

**Examples** DATA:ENCDG? might return :DATA:ENCDG SRPBINARY for the format of the outgoing waveform data.

DATA:ENCDG RPBinary sets the data encoding format to be a positive integer where the most significant byte is transferred first.

## DATA:SOURce

This command sets or returns the location of waveform data that is transferred from the instrument by the CURVe? Query.

**Group** Waveform Transfer

**Syntax** DATA:SOURce <wfm> [<,><wfm>]  
DATA:SOURce?

**Related Commands** CURVe

**Arguments** <wfm> is the location of the waveform data that will be transferred from the instrument to the controller.

**Examples** DATA:SOURCE? might return :DATA:SOURCE REF3 indicating that the source for the waveform data which is transferred using a CURVe? query is reference 3.

DATA:SOURCE CH1 specifies that the CH1 waveforms will be transferred in the next CURVe? query.

## DATA:START

This command sets or returns the starting data point for waveform transfer. This command allows for the transfer of partial waveforms to and from the instrument.

**Group** Waveform Transfer

**Syntax** DATA:START <NR1>  
DATA:START?

**Related Commands** [CURVe](#), [DATA](#), [DATA:STOP](#), [SAVe:WAVEform](#), [SAVe:WAVEform:FILEFormat](#), [WFMinpre:NR\\_Pt](#), [WFMOuppre:NR\\_Pt?](#)

**Arguments** <NR1> is the first data point that will be transferred, which ranges from 1 to the record length. Data will be transferred from <NR1> to DATA:STOP or the record length, whichever is less. If <NR1> is greater than the record length, the last data point in the record is transferred.

DATA:START and DATA:STOP are order independent. When DATA:STOP is greater than DATA:START, the values will be swapped internally for the CURVE? query.

**Examples** DATA:START? might return :DATA:START 214 indicating that data point 214 is the first waveform data point that will be transferred.

DATA:START 10 specifies that the waveform transfer will begin with data point 10.

## DATA:STOP

This command sets or returns the last data point that will be transferred when using the [CURVe?](#) query. This command allows for the transfer of partial waveforms to the controller.

Changes to the record length value are not automatically reflected in the data:stop value. As record length is varied, the data:stop value must be explicitly changed to ensure the entire record is transmitted. In other words, curve results will not automatically and correctly reflect increases in record length if the distance from data:start to data:stop stays smaller than the increased record length.

---

**NOTE.** When using the [CURVe](#) command, DATA:STOP is ignored and [WFMinpre:NR\\_Pt](#) is used.

---

<b>Group</b>	Waveform Transfer
<b>Syntax</b>	DATA:STOP <NR1> DATA:STOP?
<b>Related Commands</b>	<a href="#">CURVe</a> , <a href="#">DATA</a> , <a href="#">DATA:START</a> , <a href="#">SAVe:WAVEform</a> , <a href="#">SAVe:WAVEform:FILEFormat</a> , <a href="#">WFMinpre:NR_Pt</a> , <a href="#">WFMOuppre:NR_Pt?</a>
<b>Arguments</b>	<p>&lt;NR1&gt; is the last data point that will be transferred, which ranges from 1 to the record length. If &lt;NR1&gt; is greater than the record length, then data will be transferred up to the record length. If both DATA:START and DATA:STOP are greater than the record length, the last data point in the record is returned.</p> <p>DATA:START and DATA:STOP are order independent. When DATA:STOP is less than DATA:START, the values will be swapped internally for the CURVE? query.</p> <p>If you always want to transfer complete waveforms, set DATA:START to 1 and DATA:STOP to the maximum record length, or larger.</p>
<b>Examples</b>	<p>DATA:STOP? might return :DATA:STOP 14900 indicating that 14900 is the last waveform data point that will be transferred.</p> <p>DATA:STOP 15000 specifies that the waveform transfer will stop at data point 15000.</p>

## DATE

This command sets or returns the date that the instrument can display. This command is equivalent to selecting Set Date & Time from the Utilities menu and then setting the fields in the Date group box.

<b>Group</b>	Miscellaneous
<b>Syntax</b>	DATE <QString> DATE?
<b>Related Commands</b>	<a href="#">TIME</a>
<b>Arguments</b>	<p>&lt;QString&gt; is a date in the form "yyyy-mm-dd" where yyyy refers to a four-digit year number, mm refers to a two-digit month number from 01 to 12, and dd refers to a two-digit day number in the month.</p>

**Examples**     `DATE "2000-01-24"` specifies that the date is set to January 24, 2000.

`DATE?` might return `:DATE 2000-01-24` indicating the current date is set to January 24, 2000.

## \*DDT

This command allows you to specify a command or a list of commands that are executed when the instrument receives a [\\*TRG](#) command or the GET GPIB interface message. Define Device Trigger ([\\*DDT](#)) is a special alias that the [\\*TRG](#) command uses.

**Group**     Miscellaneous

**Syntax**     `*DDT {<Block>|<QString>}`  
`*DDT?`

**Related Commands**     [ALias](#), [\\*TRG](#)

**Arguments**     `<Block>` is a complete sequence of program messages. The messages can contain only valid commands that must be separated by semicolons and must follow all rules for concatenating commands. The sequence must be less than or equal to 80 characters. The format of this argument is always returned as a query.

`<QString>` is a complete sequence of program messages. The messages can contain only valid commands that must be separated by semicolons and must follow all rules for concatenating commands. The sequence must be less than or equal to 80 characters.

**Examples**     `*DDT #OACQUIRE:STATE RUN` specifies that the acquisition system will be started each time a [\\*TRG](#) command is sent.

## DESE

This command sets and queries the bits in the Device Event Status Enable Register (DESER). The DESER is the mask that determines whether events are reported to the Standard Event Status Register (SESR), and entered into the Event Queue. For a more detailed discussion of the use of these registers, see [Registers](#).

**Group**     Status and Error

<b>Syntax</b>	DESE <NR1> DESE?
<b>Related Commands</b>	<a href="#">*CLS</a> , <a href="#">*ESE</a> , <a href="#">*ESR?</a> , <a href="#">EVENT?</a> , <a href="#">EVMsg?</a> , <a href="#">*SRE</a> , <a href="#">*STB?</a>
<b>Arguments</b>	<p>&lt;NR1&gt; The binary bits of the DESER are set according to this value, which ranges from 1 through 255. For example, DESE 209 sets the DESER to the binary value 11010001 (that is, the most significant bit in the register is set to 1, the next most significant bit to 1, the next bit to 0, etc.).</p> <p>The power-on default for DESER is all bits set if *PSC is 1. If *PSC is 0, the DESER maintains the previous power cycle value through the current power cycle.</p> <hr/> <p><b>NOTE.</b> <i>Setting the DESER and ESER to the same value allows only those codes to be entered into the Event Queue and summarized on the ESB bit (bit 5) of the Status Byte Register. Use the <a href="#">*ESE</a> command to set the ESER.</i></p> <hr/>
<b>Examples</b>	<p>DESE 209 sets the DESER to binary 11010001, which enables the PON, URQ, EXE and OPC bits.</p> <p>DESE? might return :DESE 186, showing that the DESER contains the binary value 10111010.</p>

## DIAG:LOOP:OPTion (No Query Form)

Sets the diagnostic loop option.

<b>Group</b>	Calibration and Diagnostic
<b>Syntax</b>	DIAG:LOOP:OPTion {ALWAYS FAIL ONFAIL ONCE}
<b>Arguments</b>	<p>ALWAYS continues looping until diagnostics are stopped via the front panel or by an instrument command.</p> <p>FAIL causes looping until the first diagnostic failure or until diagnostics are stopped.</p> <p>ONFAIL causes looping on a specific test group as long as a FAIL status is returned from the test.</p> <p>ONCE executes diagnostics test sequence once.</p>
<b>Examples</b>	DIAG:LOOP:OPTION FAIL stops the diagnostic loop after the first failure.

## DIAG:LOOP:OPTion:NTIMes

Sets the diagnostic loop option to run N times.

**Group** Calibration and Diagnostic

**Syntax**    `DIAG:LOOP:OPTion:NTIMes <NR1>`  
             `DIAG:LOOP:OPTion:NTIMes?`

**Arguments**    `<NR1>` is the number of diagnostic loops.

**Examples**    `DIAG:LOOP:OPTION:NTIMES 3` sets the diagnostic loop to run three times.  
  
`DIAG:LOOP:OPTION:NTIMES?` might return `:DIAG:LOOP:OPTION:NTIMES 5`, indicating the diagnostic loop is set to run five times.

## DIAG:LOOP:STOP (No Query Form)

Stops diagnostic at the end of the current loop.

**Group** Calibration and Diagnostic

**Syntax**    `DIAG:LOOP:STOP`

**Examples**    `DIAG:LOOP:STOP` stops diagnostic at the end of the current loop.

## DIAG:MODE

Select the test suite of diagnostics that is to be run.

**Group** Calibration and Diagnostic

**Syntax**    `DIAG:MODE {POST|EXTENDED|RTC|SERVICE}`  
             `DIAG:MODE?`

## DIAG:RESUlt:FLAg? (Query Only)

Returns the pass/fail status from the last diagnostic test sequence execution. The [DIAG:RESUlt:LOG?](#) query can be used to determine which test(s) has failed.

**Group** Calibration and Diagnostic

**Syntax** `DIAG:RESUlt:FLAg?`

**Related Commands** [DIAG:RESUlt:LOG?](#)

**Returns** PASS indicating that all of the selected diagnostic tests have passed.  
FAIL indicating that at least one of the selected diagnostic tests has failed.

**Examples** `DIAG:RESULT:FLAG?` returns either `DIAG:RESULT:FLAG PASS` or `FAIL`.

## DIAG:RESUlt:LOG? (Query Only)

Returns the internal results log from the last diagnostic test sequence execution. The list contains all modules and module interfaces that were tested along with the pass/fail status of each.

**Group** Calibration and Diagnostic

**Syntax** `DIAG:RESUlt:LOG?`

**Related Commands** [DIAG:RESUlt:FLAg?](#)

**Returns** <QString> in the following format:  
<Status>--<Module name>[,<Status>--<Module name>...]

**Examples** `DIAG:RESULT:LOG?` might return `DIAG:RESULT:LOG`  
"pass--CPU,pass--Display,pass--Front  
Panel,pass--IO,pass--Acquisition,pass--FDD,pass--Application  
Key, pass--ROM".



## DIAG:SElect:ALL (No Query Form)

This command (no query form) selects all available diagnostics. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then choosing ALL from the Subsystem, Area and Test pull-down lists.

**Group** Calibration and Diagnostic

**Syntax** DIAG:SElect:ALL

**Examples** DIAG:SELECT:ALL selects all available diagnostics.

## DIAG:SElect:<function> (No Query Form)

Sets the oscilloscope to run diagnostics on the specified system function.

**Group** Calibration and Diagnostic

**Syntax** DIAG:SElect:<function>

**Arguments** <function> specifies a single oscilloscope function on which to run diagnostics. Valid values are:

ACQ test the acquisition system.

APPKey tests the application keys.

CPU tests the CPU.

DISPlay tests the display.

FPAnel tests the front panel controls.

IO tests the IO ports.

ROM tests the system read only memory.

**Examples** DIAG:SELECT:CPU sets the oscilloscope to run just CPU tests.

## DIAG:STATE (No Query Form)

This command (no query form) changes the oscilloscope operating state. Depending on the argument, diagnostics capabilities are either turned on or off.

This command is equivalent to opening the DIAG:STATE dialog (ON) or closing it (OFF).

**Group** Calibration and Diagnostic

**Syntax** `DIAG:STATE {ON|OFF}`

**Arguments** ON puts the instrument into the state in which diagnostics can be run.  
OFF disables diagnostics capabilities and returns the oscilloscope to a normal operating state.

**Examples** `DIAG:STATE OFF` turns off diagnostics capabilities and returns the instrument to a normal operating state.

## DISplay? (Query Only)

This query-only command returns the current Display settings. This command is equivalent to selecting Display Setup from the Display menu and then viewing the contents of each tab.

**Group** Display

**Syntax** `DISplay?`

**Examples** `DISPLAY?` might return `:DISPLAY:CLOCK 1;COLOR:PALETTE:IMAGEVIEW MONOGREEN; RECORDVIEW TEMPERATURE;USER:GRATICULE 165,50,15;CH1 180,50,100; CH2 300,50,100;CH3 60,50,100;CH4 240,50,100;REF1 0,90,0; REF2 0,90,100;REF3 60,90,100;REF4 240,90,100;MATH1 160,50,100; MATH2 40,60,100;MATH3 120,60,100;MATH4 195,50,100; HISTOGRAM 320,50,100;CARET 150,50,100;MASK 0,25,75;MASKHIGHLIGHT 140,50,100;:DISPLAY:COLOR:MATHCOLOR DEFAULT;REFCOLOR DEFAULT; :DISPLAY:FILTER SINX;FORMAT YT;GRATICULE IRE;INTENSITY :WAVEFORM:IMAGEVIEW 81.0000;RECORDVIEW 81.0000;:DISPLAY :INTENSITY:AUTOBRIGHT 0;SCREENSAVER 1;SCREENSAVERDELAY 28800; :DISPLAY:PERSISTENCE OFF;STYLE DOTS;TRIGBAR OFF;TRIGT 1; CURSORTICK LONG;VARPERSIST 2.6000;SCREENTEXT:STATE 1; LABEL1:NAME "";XPOS 100;YPOS 5;:DISPLAY:SCREENTEXT :LABEL2:NAME "THIS IS SCREEN TEXT";XPOS 100;YPOS 20; :DISPLAY:SCREENTEXT:LABEL3:NAME "";XPOS 100;YPOS 35;`

```
:DISPLAY:SCREENTEXT:LABEL4:NAME "";XPOS 100;YPOS 50;  
:DISPLAY:SCREENTEXT:LABEL5:NAME "";XPOS 100;YPOS 343;  
:DISPLAY:SCREENTEXT:LABEL6:NAME "";XPOS 100;YPOS 80;  
:DISPLAY:SCREENTEXT:LABEL7:NAME "";XPOS 100;YPOS 95;  
:DISPLAY:SCREENTEXT:LABEL8:NAME "";XPOS 100;YPOS 110;  
:DISPLAY:WAVEFORM 1
```

## DISplay:CLOCK

This command sets or returns the display of the date and time. This is equivalent to selecting Display Date & Time from the Display menu. The query form of this command returns an ON (1) or an OFF (0).

**Group** Display

**Syntax** `DISplay:CLOCK {ON|OFF|<NR1>}`  
`DISplay:CLOCK?`

**Related Commands** [DATE](#), [TIME](#)

**Arguments** ON enables display of date and time.  
OFF disables display of date and time.  
<NR1> = 0 disables display of date and time; any other value enables display of date and time.

**Examples** `DISPLAY:CLOCK ON` enables display of date and time.  
`DISPLAY:CLOCK?` might return `:DISPLAY:CLOCK 1` indicating that the display of date and time is currently enabled.

## DISplay:FORMat (Query Only)

This command queries the display format, and always returns YT..

**Group** Display

**Syntax** `DISplay:FORMat?`

**Arguments** YT sets the display to a voltage versus time format and is the default mode.

**Examples**     `DISPLAY:FORMAT?` returns `:DISPLAY:FORMAT YT` is for the display format, indicating that the display is in the YT mode.

## DISplay:GRAticule

This command selects or queries the type of graticule that is displayed. This command is equivalent to selecting Graticule Style from the Display menu.

**Group**     Display

**Syntax**     `DISplay:GRAticule {CROSSHair|FRaMe|FULl|GRId}`  
`DISplay:GRAticule?`

**Arguments**     `CROSSHair` specifies a frame and cross hairs.  
`FRaMe` specifies a frame only.  
`FULl` specifies a frame, a grid and cross hairs.  
`GRId` specifies a frame and grid only.

**Examples**     `DISPLAY:GRATICULE FRaMe` sets the graticule type to display the frame only.  
`DISPLAY:GRATICULE?` might return `:DISPLAY:GRATICULE FULL` indicating that all graticule elements are selected.

## DISplay:INTENSITy? (Query Only)

This query-only command returns the waveform saturation level, autobright state, and screen saver settings. This command is equivalent to selecting Display Setup from the Display menu and choosing the Appearance tab.

**Group**     Display

**Syntax**     `DISplay:INTENSITy?`

**Arguments**     None

**Examples**     `DISPLAY:INTENSITY?` might return  
                  `:DISPLAY:INTENSITY:WAVEFORM:IMAGEVIEW`  
                  `75.0000;RECORDVIEW 10.0000;DISPLAY:INTENSITY:AUTOBRIGHT`  
                  `1;SCREENSAVER 1;SCREENSAVERDELAY 28800`

## DISplay:INTENSITy:BACKLight

Sets or returns the backlight intensity for the display. This is equivalent to setting Backlight Intensity in the **DISPLAY** menu.

**Group**     Display

**Syntax**     `DISplay:INTENSITY:BACKLight {LOW|MEDIum|HIGH}`  
                  `DISplay:INTENSITY:BACKLight?`

**Examples**     `DISPLAY:INTENSITY:BACKLIGHT?` might return  
                  `DISPLAY:INTENSITY:BACKLIGHT HIGH`

## DISplay:INTENSITy:GRAticule

Sets or returns the graticule intensity for the display.

**Group**     Display

**Syntax**     `DISplay:INTENSITY:GRAticule <NR1>`  
                  `DISplay:INTENSITY:GRAticule?`

**Arguments**     `<NR1>` is the graticule intensity and ranges from 1 to 100 percent.

**Examples**     `DISPLAY:INTENSITY:GRATICULE?` might return  
                  `DISPLAY:INTENSITY:GRATICULE 30`

## DISplay:INTENSITy:WAVEform

Sets or returns the intensity of the waveforms. This command is equivalent to adjusting the **WAVEFORM INTENSITY** knob.

**Group**     Display

**Syntax**     `DISplay:INTENSITY:WAVEform <NR1>`  
`DISplay:INTENSITY:WAVEform?`

**Arguments**     `<NR1>` is the waveform intensity and ranges from 1 to 100 percent.

**Examples**     `DISPLAY:INTENSITY:WAVEFORM?` might return  
`DISPLAY:INTENSITY:WAVEFORM 60`  
as the intensity of the waveforms.

## DISplay:PERsistence

This command sets or returns the persistence aspect of the display. This affects the display only and is equivalent to selecting Display Persistence from the Display menu.

**Group**     Display

**Syntax**     `DISplay:PERsistence {CLEAR|AUTO|MINImum|INFInite}`  
`DISplay:PERsistence?`

**Arguments**     `CLEAR` specifies the time, in seconds, that waveform points are displayed on the screen.

`INFInite` displays waveform points until a control change resets the acquisition system.

`AUTO` specifies that the waveform persistence is controlled by the front panel `WAVEFORM INTENSITY` knob.

`MINImum` specifies that the waveform persistence is set to the minimum value of `0.0E0`.

**Examples**     `DISPLAY:PERSISTENCE VARPersist` sets the persistence aspect of the display to fade.

`DISPLAY:PERSISTENCE?` might return `:DISPLAY:PERSISTENCE OFF` indicating that the persistence aspect of the display is disabled.

## DISplay:PICTure:AUTOContrast

Sets or returns the video picture mode autocontrast setting.

<b>Conditions</b>	Requires DPO40VID application module.
<b>Group</b>	Display
<b>Syntax</b>	<code>DISplay:PICTure:AUTOContrast {OFF ON &lt;NR1&gt;}</code> <code>DISplay:PICTure:AUTOContrast?</code>
<b>Related Commands</b>	<a href="#">DISplay:PICTure:BRiGhtness</a> , <a href="#">DISplay:PICTure:CONTRAst</a> , <a href="#">DISplay:PICTure:STATE</a>
<b>Arguments</b>	OFF or <NR1> = 0 disables picture autocontrast mode. ON or <NR1> ≠ 0 enables picture autocontrast mode.
<b>Examples</b>	DISPLAY:PICTURE:AUTOCONTRAST? might return DISPLAY:PICTURE:AUTOCONTRAST ON, indicating that the video picture autocontrast mode is enabled.

## DISplay:PICTure:BRiGhtness

Sets or returns the video picture mode brightness setting.

<b>Conditions</b>	Requires DPO40VID application module.
<b>Group</b>	Display
<b>Syntax</b>	<code>DISplay:PICTure:BRiGhtness &lt;NR1&gt;</code> <code>DISplay:PICTure:BRiGhtness?</code>
<b>Related Commands</b>	<a href="#">DISplay:PICTure:AUTOContrast</a> , <a href="#">DISplay:PICTure:CONTRAst</a> , <a href="#">DISplay:PICTure:STATE</a>
<b>Arguments</b>	<NR1> sets the picture brightness value from 0 to 100. DISplay:PICTure:AUTOContrast must be OFF in order to set the brightness value.
<b>Examples</b>	DISPLAY:PICTURE:BRIGHTNESS? might return DISPLAY:PICTURE:BRIGHTNESS 34, indicating that the video picture brightness setting is 34.

## DISplay:PICTure:CONTRast

Sets or returns the video picture mode contrast setting.

<b>Conditions</b>	Requires DPO40VID application module.
<b>Group</b>	Display
<b>Syntax</b>	DISplay:PICTure:CONTRast <NR1> DISplay:PICTure:CONTRast?
<b>Related Commands</b>	<a href="#">DISplay:PICTure:AUTOContrast</a> , <a href="#">DISplay:PICTure:BRiGhtness</a> , <a href="#">DISplay:PICTure:STATE</a>
<b>Arguments</b>	<NR1> sets the picture contrast value from 0 to 100. DISplay:PICTure:AUTOContrast must be OFF in order to set the contrast value.
<b>Examples</b>	DISPLAY:PICTURE:CONTRAST? might return DISPLAY:PICTURE:CONTRAST 45, indicating that the video picture contrast setting is 45.

## DISplay:PICTure:STATE

Sets or returns the video picture mode setting.

<b>Conditions</b>	Requires DPO40VID application module.
<b>Group</b>	Display
<b>Syntax</b>	DISplay:PICTure:STATE <NR1> DISplay:PICTure:STATE?
<b>Related Commands</b>	<a href="#">DISplay:PICTure:BRiGhtness</a> , <a href="#">DISplay:PICTure:CONTRast</a> , <a href="#">DISplay:PICTure:AUTOContrast</a>
<b>Arguments</b>	OFF or <NR1> = 0 disables video picture mode. ON or <NR1> ≠ 0 enables video picture mode.



**Examples**     `DISPLAY:PICTURE:STATE` might return `DISPLAY:PICTURE:STATE ON`, indicating that the video picture mode is enabled.

## DISplay:STYle:DOTsonly

This command sets a dots-only display.

**Group**     Display

**Syntax**     `DISplay:STYle:DOTsonly {<NR1>|OFF|ON}`  
`DISplay:STYle:DOTsonly?`

**Arguments**     ON or `<NR1> ≠ 0` turns on the dots-only display.  
OFF or `<NR1> = 0` turns off the dots-only display.

## DISplay:VECtroscope:STATE

Sets or returns the video vectorscope state setting. This is equivalent to turning Vectorscope mode on or off from the front panel.

**Conditions**     This command requires a DPO4VID application module.

**Group**     Display

**Syntax**     `DISplay:VECtroscope:STATE {ON|OFF|<NR1>}`  
`DISplay:VECtroscope:STATE?`

**Arguments**     ON or `<NR1> = 1` enables the vectorscope display mode.  
OFF or `<NR1> = 0` disables the vectorscope display mode.

**Examples**     `DISPLAY:VECTROSCOPE:STATE` might return `DISPLAY:VECTROSCOPE:STATE 0` indicating that the vectorscope display mode is disabled (off).

## DISplay:VECtroscope:STsetting

**Conditions**     This command requires a DPO4VID application module.

<b>Group</b>	Display
<b>Syntax</b>	DISplay:VEctroscope:STsetting DISplay:VEctroscope:STsetting?

## DISplay:VEctroscope:TYPE

Sets or returns the video vectorscope display type setting.

**Conditions** This command requires a DPO4VID application module.

**Group** Display

**Syntax** DISplay:VEctroscope:TYPE {ONEHundred|SEVENTYFive|<NR1>}  
DISplay:VEctroscope:TYPE?

**Arguments** ONEHundred or <NR1>  $\geq 1$  sets the vectorscope display mode to 100% colorbars.  
SEVENTYFive or <NR1>  $\leq 0$  sets the vectorscope display mode to 75% colorbars.

**Examples** DISPLAY:VECTROSCOPE:TYPE? might return DISPLAY:VECTROSCOPE:TYPE  
1 indicating that the vectorscope display mode is set to 100%.

## DISplay:VEctroscope:TYsetting

Sets or returns the video vectorscope display type setting state. Note that this does not actually turn on the vectorscope display type, but just sets the type value.

**Conditions** This command requires a DPO4VID application module.

**Group** Display

**Syntax** DISplay:VEctroscope:TYsetting {ONEHundred|SEVENTYFive|<NR1>}  
DISplay:VEctroscope:TYsetting?

**Arguments** ONEHundred or <NR1>  $\geq 1$  sets the vectorscope display mode to 100%.  
SEVENTYFive or <NR1>  $\leq 0$  sets the vectorscope display mode to 75%.

**Examples** `DISPLAY:VECTORSCOPE:TYSETTING?` might return `DISPLAY:VECTORSCOPE:TY_SETTING 1` indicating that the vectorscope display mode is set to 100%.

## DISplay:XY:MODe

Sets or returns the display mode. This command is equivalent to setting XY Display in the **DISPLAY** menu.

**Group** Display

**Syntax** `DISplay:XY:MODe {OFF|TRIGger|GATEDxyz|VEctorscope}`  
`DISplay:XY:MODe?`

**Related Commands** [DISplay:XY:YCHannel](#), [DISplay:XY:YREF](#), , , [DISplay:FORMat](#)

**Arguments** `OFF` sets the display to a voltage versus time format and is the normal mode.

`TRIGger` displays the voltage of one waveform against the voltage of another. The sources that make up an XY waveform are predefined and are listed in Table 2-10. Displaying one source causes its corresponding source to be displayed.

**Table 2-33: XY format pairs**

X-Axis source	Y-Axis source
Ch1	Ch2, Ch3, or Ch4
Ref1	Ref2, Ref3, or Ref4

`GATEDxyz` displays XY signals only when the Z (gating) channel is true. Gated XYZ is similar to analog oscilloscope modulated XYZ mode except that the displayed XY signal is either on or off; there is no intensity modulation. Gated XYZ is useful for showing constellation diagrams. Use the `DISplay:XY:ZCHannel` and `DISplay:XY:ZLEVel` commands to specify the Z channel and Z channel threshold level.

`VEctorscope` sets the display to a video vectorscope display mode.

**Examples** `DISPLAY:XY:MODE OFF` selects a voltage versus time format for the display.

`DISPLAY:XY:MODE?` might return `DISPLAY:XY:MODE TRIGGER` indicating a triggered XY display format.

## DISplay:XY:YCHannel

Specifies the Y channel to be displayed with channel 1 in Triggered XY display mode.

**Group** Display

**Syntax** DISplay:XY:YCHannel {CH2|CH3|CH4|OFF}  
DISplay:XY:YCHannel?

**Related Commands** [DISplay:XY:MODE](#), [DISplay:XY:YREF](#),

**Arguments** CH2, CH3, and CH4 specify the channel displayed with channel 1 in Triggered XY Display mode.

OFF sets the Y Channel to off or none which turns off or prevents the CH1 versus CH <x> waveform from being displayed in Triggered XY or prevents it from coming on if Triggered XY is turned on later.

**Examples** DISPLAY:XY:YCHANNEL CH2 sets channel 2 to be displayed with channel 1 in Triggered XY mode.

DISPLAY:XY:YCHANNEL? might return DISPLAY:XY:YCHANNEL CH2 indicating that channel 2 will be the Y channel in Triggered XY mode.

## DISplay:XY:YREF

Specifies the Y channel to be displayed with reference 1 in Triggered XY display mode.

**Group** Display

**Syntax** DISplay:XY:YREF?

**Related Commands** [DISplay:XY:MODE](#), [DISplay:XY:YCHannel](#)

**Arguments** REF2, REF3, and REF4 specify the reference displayed with REF1 in Triggered XY Display mode.

OFF set the Y reference to off which turns off or prevents the REF1 versus REF<x> waveform from being displayed in Triggered XY or prevents it from coming on if Triggered XY is turned on later.

**Examples**     `DISPLAY:XY:YREF REF2` sets REF2 to be displayed with REF1 in Triggered XY mode.

## \*ESE

This command sets and queries the bits in the Event Status Enable Register (ESER). The ESER prevents events from being reported to the Status Byte Register (STB). For a more detailed discussion of the use of these registers, see Registers.

**Group**     Status and Error

**Syntax**     `*ESE <NR1>`  
`*ESE?`

**Related Commands**     [\\*CLS](#), [DESE](#), [\\*ESR?](#), [EVENT?](#), [EVMsg?](#), [\\*SRE](#), [\\*STB?](#)

**Arguments**     <NR1> specifies the binary bits of the ESER according to this value, which ranges from 0 through 255.

The power-on default for the ESER is 0 if \*PSC is 1. If \*PSC is 0, the ESER maintains the previous power cycle value through the current power cycle.

---

**NOTE.** *Setting the DESE and the ESER to the same values allows only those codes to be entered into the Event Queue and summarized on the ESB bit (bit 5) of the Status Byte Register. Use the DESE command to set the DESE.*

---

**Examples**     `*ESE 209` sets the ESER to binary 11010001, which enables the PON, URQ, EXE, and OPC bits.

`*ESE?` might return 186, showing that the ESER contains the binary value 10111010.

## \*ESR? (Query Only)

This query-only command returns the contents of the Standard Event Status Register (SESR). \*ESR? also clears the SESR (since reading the SESR clears it). For a more detailed discussion of the use of these registers, see Registers.

**Group** Status and Error

**Syntax** \*ESR?

**Related Commands** [ALLEv?](#), [\\*CLS](#), [DESE](#), [\\*ESE](#), [EVENT?](#), [EVMsg?](#), [\\*SRE](#), [\\*STB?](#)

**Examples** \*ESR? might return 213, showing that the SESR contains the binary value 11010101.

## ETHERnet:DHCPbootp

Sets or returns the network initialization search for a DHCP/BOOTP server.

**Group** Ethernet

**Syntax** ETHERnet:DHCPbootp {ON|OFF|<NR1>}  
ETHERnet:DHCPbootp?

**Arguments** ON or <NR1>  $\geq 1$  enables the oscilloscope to search the network for a DHCP or BOOTP server in order to automatically assign a dynamic IP address to the oscilloscope.

---

**NOTE.** Do not use DHCP/BOOTP searching if your oscilloscope has been assigned a static address on a network. If you set this command to ON, the DHCP/BOOTP search will delete or change your static IP address information.

---

OFF or <NR1>  $\leq 0$  disables the oscilloscope to search the network for a DHCP or BOOTP server.

**Examples** ETHERNET:DHCPBOOTP ON sets the oscilloscope to search for a DHCP or BOOTP server and assign a dynamic IP address to the oscilloscope.

## ETHERnet:DNS:IPADdress

Sets or returns the network Domain Name Server (DNS) IP address.

**Group** Ethernet

**Syntax** ETHERnet:DNS:IPADdress <QString>  
ETHERnet:DNS:IPADdress?

**Arguments** <QString> is a standard IP address value, enclosed in quotes.

**Examples** ETHERNET:DNS:IPADDRESS "128.196.13.352" sets the DNS IP address that the oscilloscope uses to communicate with the network.

## ETHERnet:DOMAINname

Sets or returns the network domain name.

**Group** Ethernet

**Syntax** ETHERnet:DOMAINname <Qstring>  
ETHERnet:DOMAINname?

**Arguments** <QString> is the network domain name, enclosed in quotes.

**Examples** ETHERNET:DOMAINNAME "Alpha1.Mycorp.com" sets the domain name that the oscilloscope uses to communicate with the network.

## ETHERnet:ENET:ADdress? (Query Only)

Returns the Ethernet address value assigned to the oscilloscope. This is assigned at the factory and can not be changed.

**Group** Ethernet

**Syntax** ETHERnet:ENET:ADdress?

**Examples**     `ETHERNET:ENET:ADDRESS?` returns an ethernet address such as "02:00:e3:01:32:03"

## ETHERnet:GATEWay:IPADdress

This command sets or returns the remote interface gateway IP address.

**Group**     Ethernet

**Syntax**     `ETHERnet:GATEWay:IPADdress <QString>`  
`ETHERnet:GATEWay:IPADdress?`

**Arguments**     `<QString>` is a standard IP address value, enclosed in quotes.

**Examples**     `ETHERNET:GATEWAY:IPADDRESS "128.143.16.1"` sets the gateway IP address.

## ETHERnet:HTTPEndpoint

Sets or returns the remote interface HTTP port value.

**Group**     Ethernet

**Syntax**     `ETHERnet:HTTPEndpoint <QString>`  
`ETHERnet:HTTPEndpoint?`

**Arguments**     `<QString>` is an integer port number, enclosed in quotes.

**Examples**     `ETHERNET:HTTPPORT "45"` sets the HTTP port value to 45.

## ETHERnet:IPADdress

Sets or returns the IP address assigned to the oscilloscope.

**Group**     Ethernet



<b>Syntax</b>	ETHERnet:IPADDRESS <QString> ETHERnet:IPADDRESS?
<b>Arguments</b>	<QString> is a standard IP address value, enclosed in quotes.
<b>Examples</b>	ETHERNET:IPADDRESS "123.121.13.214" sets the oscilloscope's IP address.

## ETHERnet:NAME

Sets or returns the network name assigned to the oscilloscope.

<b>Group</b>	Ethernet
<b>Syntax</b>	ETHERnet:NAME <QString> ETHERnet:NAME?
<b>Arguments</b>	<QString> is the network name assigned to the oscilloscope, enclosed in quotes.
<b>Examples</b>	ETHERNET:NAME "labscope1" sets the oscilloscope's network name.

## ETHERnet:PASSWord

This command sets or returns the Ethernet access password.

<b>Group</b>	Ethernet
<b>Syntax</b>	ETHERnet:PASSWord <old>,<new> ETHERnet:PASSWord?
<b>Arguments</b>	<old> is the current password, enclosed in quotes. If there is no current password, enter a null character (two quotes with no character between).  <new> is a new password, enclosed in quotes.
<b>Examples</b>	ETHERNET:PASSWORD "karma2","ZEN53" replaces the current Ethernet password karma2 with the new password ZEN53.  ETHERNET:PASSWORD? might return :ETHERNET:PASSWORD "ZEN53".

## ETHERnet:SUBNETMask

Sets or returns the remote interface subnet mask value.

**Group** Ethernet

**Syntax** ETHERnet:SUBNETMask <QString>  
ETHERnet:SUBNETMask?

**Arguments** <QString> is the subnet mask value, enclosed in quotes.

**Examples** ETHERNET:SUBNETMASK "255.255.255.0" sets the subnet mask value using standard IP address notation format.

## EVENT? (Query Only)

This query-only command returns an event code from the Event Queue that provides information about the results of the last [\\*ESR?](#) read. EVENT? also removes the returned value from the Event Queue.

**Group** Status and Error

**Syntax** EVENT?

**Related Commands** [ALLEv?](#), [\\*CLS](#), [DESE](#), [\\*ESE](#), [\\*ESR?](#), [EVMsg?](#), [\\*SRE](#), [\\*STB?](#)

**Examples** EVENT? might return :EVENT 110, showing that there was an error in a command header.

## EVMsg? (Query Only)

This query-only command removes a single event code from the Event Queue that is associated with the results of the last [\\*ESR?](#) read and returns the event code along with an explanatory message. For more information, see Event Handling.

**Group** Status and Error

**Syntax** EVMsg?

**Related Commands**    [ALLEv?](#)  
[\\*CLS](#), [DESE](#), [\\*ESE](#), [\\*ESR?](#), [EVENT?](#), [\\*SRE](#), [\\*STB?](#)

**Returns**    The event code and message in the following format:  
                 <Event Code><Comma><QString>[<Event  
                 Code><Comma><QString>...]<QString>::= <Message>;[<Command>]  
                 where <Command> is the command that caused the error and may be returned  
                 when a command error is detected by the instrument. As much of the command  
                 will be returned as possible without exceeding the 60 character limit of the  
                 <Message> and <Command> string combined. The command string is  
                 right-justified.

**Examples**    EVMSG? might return :EVMSG 110,"Command header error".

## EVQty? (Query Only)

This query-only command returns the number of event codes that are in the Event Queue. This is useful when using the [ALLEv?](#) query, since it lets you know exactly how many events will be returned.

**Group**    Status and Error

**Syntax**    EVQty?

**Related Commands**    [ALLEv?](#), [EVENT?](#), [EVMsg?](#)

**Examples**    EVQTY? might return :EVQTY 3, indicating the number of event codes in the Event Queue.

## FACTory (No Query Form)

This command (no query form) resets the instrument to its factory default settings.

This command is equivalent to pressing the DEFAULT SETUP button located on the instrument front panel or selecting Recall Default Setup from the File menu.

This command resets the instrument settings to factory defaults:

- Clears the Event Status Enable Register.
- Clears the Service Request Enable Register.
- Sets the Device Event Status Enable Register to 255.
- Sets the Power On Status Clear Flag to TRUE.
- Purges all defined aliases.
- Enables all Command Headers.
- Sets the macro defined by \*DDT to a "zero-length field."
- Clears the pending operation flag and associated operations.

This command does not reset the following:

- The state of the GPIB (IEEE Std 488.1-1987) interface.
- The selected GPIB address.
- Calibration data that affects device specifications.
- Protected user data.
- Stored settings.
- The current password (if you set one).

**Group** Save and Recall

**Syntax** FACTory

**Related Commands** [\\*PSC](#), [\\*RCL](#), [RECALL:SETUp](#), [\\*RST](#), [\\*SAV](#), [SAVE:SETUp](#)

**Arguments** None

**Examples** FACTORY resets the instrument to its factory default settings.

## FASTAcq? (Query Only)

This query-only command returns the state of Fast Acquisitions. This command is equivalent to pressing the FASTACQ button on the front panel.

**Group** Acquisition

**Syntax** FASTAcq?

**Examples** FASTACQ? might return the following string for the current acquisition parameters:  
:FASTACQ:STATE 0

## FASTAcq:STATE

This command sets or returns the state of Fast Acquisitions. This command is equivalent to the FASTACQ button on the front panel.

**Group** Acquisition

**Syntax** FASTAcq:STATE {ON|OFF|<NR1>}  
FASTAcq:STATE?

**Arguments** ON enables Fast Acquisitions mode.  
OFF disables Fast Acquisitions mode.  
<NR1> = 0 disables Fast Acquisitions mode; any other value enables Fast Acquisitions mode.

**Examples** FASTACQ:STATE ON enables the Fast Acquisitions mode.  
FASTACQ:STATE? might return :FASTACQ:STATE 1 indicating that Fast Acquisitions mode is currently active.

## FILESystem? (Query Only)

This query-only command returns the directory listing of the current working directory. This query is the same as the FILESystem:DIR? query.

**Group** File System

**Syntax** FILESystem?

**Related Commands** [FILESystem:COpy](#), [FILESystem:CWD](#), [FILESystem:DELEte](#),  
[FILESystem:DIR?](#), [FILESystem:REName](#)

**Arguments** None.

**Examples** FILESYSTEM? might return :FILESYSTEM:DIR,  
".", "..", "myFile.txt", "myWaveform.wfm".

## FILESystem:COPy (No Query Form)

This command (no query form) copies a named file to a new file. The new file may be in a totally separate directory than the old file. You can only copy one file at a time using this command. Wild card characters are not allowed.

**Group** File System

**Syntax** FILESystem:COPy {<source file path>,<destination file path>}

**Related Commands** [FILESystem:CWD](#), [FILESystem:DELEte](#)

**Arguments** <file path> is a quoted string that defines the file name and path. If the file path is within the current working directory, you need only specify the file name.

**Examples** FILESYSTEM:COPy "C:\MYDIR\TEK00001.SET",  
"C:\ANOTHERDIR\COPYOFTEK001.SET" copies the file named  
TEK00001.SET, located in the MYDIR directory on the C drive to a file named  
COPYOFTEK001.SET in the ANOTHERDIR directory on the C drive.

## FILESystem:CWD

This command sets or returns the current working directory for FILESystem GPIB commands. The default working directory is "C:\TekScope". Anytime that you use this command to change the directory, the directory that you specify is retained as the current working directory until you either change the directory or you delete the directory. If you delete the current working directory, the instrument resets current working directory to the default directory (C:\TekScope) the next time the instrument is powered on or the next time you execute a file system command. The current working directory is retained between power cycles.

This command supports the permutations of file and directory names that are supported by Microsoft Windows:

- Relative path names; for example, ".\Temp"
- Absolute path names; for example, "C:\Windows\System"
- Implied relative path names; for example "newfile.txt" becomes "C:\TekScope\newfile.txt" if the current working directory is "C:\TekScope"

<b>Group</b>	File System
<b>Syntax</b>	FILESystem:CWD {<new working directory path>}
<b>Arguments</b>	<new working directory path> is a quoted string that defines the current working; a directory name can be up to 128 characters.
<b>Examples</b>	<p>FILESYSTEM:CWD "C:\TekScope\images" sets the current working directory to images.</p> <p>FILESYSTEM:CWD? might return :FILESYSTEM:CWD "C:\TekScope\waveforms" indicating that the current working directory is set to Waveforms.</p>

## FILESystem:DELEte (No Query Form)

This command (no query form) deletes a named file. It does not delete directories. Use the [FILESystem:RMDir](#) command to delete a directory.

<b>Group</b>	File System
<b>Syntax</b>	FILESystem:DELEte <file path>
<b>Related Commands</b>	<a href="#">FILESystem:COPy</a> , <a href="#">FILESystem:CWD</a> , <a href="#">FILESystem:RMDir</a>
<b>Arguments</b>	<file path> is a quoted string that defines the file name and path. If the file path is within the current working directory, you need only specify the file name.
<b>Examples</b>	FILESYSTEM:DELETE "NOT_MINE.SET" deletes the file named NOT_MINE.SET from the current working directory.

## FILESystem:DIR? (Query Only)

This query-only command returns a list of quoted strings. Each string contains the name of a file or directory in the current working directory.

<b>Group</b>	File System
<b>Syntax</b>	FILESystem:DIR?
<b>Related Commands</b>	<a href="#">FILESystem:CWD</a> , <a href="#">FILESystem:MKDir</a>
<b>Arguments</b>	None
<b>Examples</b>	FILESYSTEM:DIR? returns a list of files and directories in the current working directory.

## FILESystem:FREESpace? (Query Only)

This query-only command returns the number of bytes of free space on the current drive.

<b>Group</b>	File System
<b>Syntax</b>	FILESystem:FREESpace?

## FILESystem:MKDir (No Query Form)

This command (no query form) creates a new directory.

<b>Group</b>	File System
<b>Syntax</b>	FILESystem:MKDir <directory path>
<b>Related Commands</b>	<a href="#">FILESystem:CWD</a> , <a href="#">FILESystem:DIR?</a>
<b>Arguments</b>	<directory path> is a quoted string that specifies the directory to create



**Examples**     `FILESYSTEM:MKDIR "C:\NewDirectory"` creates the directory named *NewDirectory* at the root of the C drive.

These two commands create the directory *MyNewSubDirectory* within the existing directory *MyDirectory* at the root of the C drive:

```
FILESYSTEM:CWD "C:\MyDirectory"; FILESYSTEM:MKDIR
"MyNewSubDirectory"
```

This, of course, assumes that *C:\MyDirectory* already existed and was not a read-only directory.

## FILESystem:REName (No Query Form)

This command (no query form) assigns a new name to an existing file.

**Group**     File System

**Syntax**     `FILESystem:REName <old file path>,<new file path>`

**Related Commands**     [FILESystem:CWD](#)

**Arguments**     `<old file path>` is a quoted string that defines the file name and path. If the file path is within the current working directory, you need only specify the file name.

`<new file path>` is a quoted string that defines the file name and path. If the file path is within the current working directory, you need only specify the file name.

**Examples**     `FILESYSTEM:RENAME "C:\TEK00000.SET", "C:\MYSETTING.SET"` gives the file named *TEK00000.SET* the new name of *MYSETTING.SET*. The file remains in the root directory on the C drive.

## FILESystem:RMDir (No Query Form)

This command (no query form) deletes a named directory. The directory cannot contain any files. If there are files in the directory, they must first be deleted. The directory must not be a read-only directory.

**Group**     File System

**Syntax**     `FILESystem:RMDir <directory path>`

**Related Commands**     [FILESystem:CWD](#)

**Arguments**     <directory path> is a quoted string that defines the file name and path. If the file path is within the current working directory, you need only specify the file name.

**Examples**     FILESYSTEM:RMDIR "C:\OldDirectory" removes the directory named OldDirectory from the root of the C drive.

## FPANEL:PRESS (No Query Form)

Duplicates the action of pressing a specified front-panel button.

**Group**     Miscellaneous

**Syntax**     FPanel:PRESS <button>

**Arguments**     <button> is a name of a front-panel button. Most of the argument names associate directly with their front panel button. For example, AUTOSet is for the AUTOSet button. The few commands that do not have obvious associations are listed below.

**Table 2-34: FPanel:PRESS arguments**

Argument	Description
ACQuire	Acquire MENU button
BMENU<x>	Screen bottom menu buttons, where <x>=1 for the left-most bottom menu button and <x>=7 for the right-most bottom menu button
BUS<x>	Bus select buttons, where <x> = 1 or 2.
CH<x>	Channel select button, where <x>=1 for channel 1, <x>=2 for channel 2, and so on
CLEARMenu	MENU OFF button
MAGnify	MAGNIFY button
MEASUrement	MEASURE button
OFF	Vertical menu waveform channel OFF button
RMENU<x>	Screen side menu buttons, where <x>=1 for the top-most side menu button and <x>=5 for the bottom-most side menu button

**Table 2-34: FPanel:PRESS arguments, (cont.)**

Argument	Description
TRIGger	Trigger MENU button
VERTical	Vertical MENU button

**Examples**     **FPANEL :PRESS AUTOSET** executes the oscilloscope Autoset function.

## FPANEL:TURN (No Query Form)

Duplicates the action of turning a specified front-panel control knob.

**Group**     Miscellaneous

**Syntax**     **FPANEL :TURN <knob>**

**Arguments**     **<knob>** is the name for a rotating control. Most of the argument names associate directly with their front panel knob. For example, GPKNOB is for the general purpose knob, HORZSCALE is for the horizontal scale knob, and so on.

, (comma) separates the control knob argument from the numeric rotation value argument. You do not need a white space between the arguments and the comma.

**<n>** represents the rotation direction and magnitude of rotation. Negative values represent a counterclockwise knob rotation, and positive values represent a clockwise rotation. The magnitude of **<n>** specifies the amount of the turn, where **<n> = 1** represents turning the knob one unit, **<n> = 2** represents turning the knob two units, **<n> = 5** represents turning the knob five units, and so on. The range of units depends on which front panel knob is specified.

**Examples**     **FPANEL :TURN TRIGLEVEL ,10** duplicates turning the front panel Trigger LEVEL knob clockwise by 10 units.

## GPIBUsb:ADdResS? (Query Only)

This command returns the GPIB/USB address.

**Group**     Miscellaneous

**Syntax**     **GPIBUsb:ADdResS?**

## **GPIBUsb:HWVersion**

This command sets or returns the GPIB/USB hardware version.

**Group**     Miscellaneous

**Syntax**     GPIBUsb:HWVersion  
                GPIBUsb:HWVersion?

## **GPIBUsb:ID? (Query Only)**

This command sets or returns the GPIB/USB ID.

**Group**     Miscellaneous

**Syntax**     GPIBUsb:ID?

## **GPIBUsb:SETADdress (No Query Form)**

**Group**     Miscellaneous

**Syntax**     GPIBUsb:SETADdress

## **GPIBUsb:SETID (No Query Form)**

**Group**     Miscellaneous

**Syntax**     GPIBUsb:SETID

## **GPIBUsb:STATUS**

**Group**     Miscellaneous

**Syntax**     GPIBUsb:STATUS  
                GPIBUsb:STATUS?

## HARDCopy (No Query Form)

This command sends a copy of the screen display to the active printer using the current palette and layout settings.

**Group** Hard Copy

**Syntax** HARDCopy

**Related Commands** [\\*WAI](#), [\\*CLS](#)

**Examples** HARDCOPY initiates a screen copy to the active system printer.

## HARDCopy:ACTIVEprinter

Sets or returns the currently active printer. When a hardcopy operation is performed, the output will be sent to this printer. One of two methods of specifying the printer can be used: an index value obtained from looking at the list of attached printers, or by specifying the printer name.

**Group** Hard Copy

**Syntax** HARDCopy:ACTIVEprinter {<NR1>|<name>}  
HARDCopy:ACTIVEprinter?

**Arguments** <NR1> is the index of the desired printer as returned from [HARDCopy:PRINTER:LIST?](#)

<name> is the name of the printer as specified in the printer list. This name is case sensitive and must be entered exactly as shown in the list.

## HARDCopy:INKSaver

Changes hard copy output to print color traces and graticule on a white background while retaining waveform color information (except for channel 1, which prints a dark blue because yellow does not show up well is difficult to see on a white background). This option can significantly reduce print time and quantities of ink required compared with WYSIWYG dark background images. Returns 1 when inksaver is on and 0 when inksaver is turned off.

<b>Group</b>	Hard Copy
<b>Syntax</b>	<code>HARDCopy:INKSaver?</code>
<b>Arguments</b>	<code>ON</code> or <code>&lt;NR1&gt; ≠ 0</code> sets the ink saver mode on. <code>OFF</code> or <code>&lt;NR1&gt; = 0</code> sets the ink saver mode off.
<b>Examples</b>	<code>HARDCOPY:INKSAVER ON</code> will cause subsequent hard copy output and preview commands to display grayscale menus on a white background.

## **HARDCopy:LAYout**

This command sets or returns the page orientation for hard copy.

<b>Group</b>	Hard Copy
<b>Syntax</b>	<code>HARDCopy:LAYout {PORTRait LANDscape}</code> <code>HARDCopy:LAYout?</code>
<b>Arguments</b>	<code>PORTRait</code> orients the screen image vertically on the printed page. <code>LANDscape</code> orients the screen image horizontally on the printed page.
<b>Examples</b>	<code>HARDCOPY:LAYOUT LANDSCAPE</code> sets the hard copy page orientation to Landscape. <code>HARDCOPY:LAYOUT?</code> might return <code>:HARDCOPY:LAYOUT PORTRAIT</code> indicating that the hard copy page orientation is set to portrait.

## **HARDCopy:PREVIEW (No Query Form)**

Causes a preview of the current screen contents with the InkSaver palette applied to be displayed.

<b>Group</b>	Hard Copy
<b>Syntax</b>	<code>HARDCopy:PREVIEW {ON OFF &lt;NR1&gt;}</code>

**Arguments**    ON or <NR1> ≠ 0 turns preview mode On.  
                  OFF or <NR1> = 0 turns preview mode Off.

## **HARDCopy:PRINTer:ADD (No Query Form)**

Adds a network printer to the list of available printers. All three arguments must be present, but only one of server name or server IP address must be specified. An empty string can be used for blank arguments.

**Group**        Hard Copy

**Syntax**       HARDCopy:PRINTer:ADD <name>,<server>,<address>

**Arguments**    <name> - The name of the network printer queue.  
                  <server> - The host name of the print (LPR) server.  
                  <address> - The IP address of the print server.

## **HARDCopy:PRINTer:DELeTe (No Query Form)**

Removes a network printer from the list of available printers. Either the index from [HARDCopy:PRINTer:LIST?](#) or the printer name can be used as an argument. The printer name is case-sensitive.

**Group**        Hard Copy

**Syntax**       HARDCopy:PRINTer:DELeTe {<index>|<name>}

**Arguments**    <index> is the index of the printer to be deleted  
                  <name> is the name of the printer to be deleted.

## **HARDCopy:PRINTer:LIST? (Query Only)**

Returns the list of currently attached printers.

**Group**        Hard Copy

**Syntax**     `HARDCopy:PRINter:LIST?`

## **HARDCopy:PRINter:REName (No Query Form)**

Renames a network printer on the list of available printers, replacing the currently stored settings with the settings specified in the command. Either the index from [HARDCopy:PRINter:LIST?](#) or the printer name can be used as the first argument. Four arguments must be present, but the arguments may be empty strings if the value for a field is to be deleted.

**Group**     Hard Copy

**Syntax**     `HARDCopy:PRINter:REName  
{<index>|<name>},<new_name>,<new_server>,<new_address>`

**Arguments**     `<index>` is the index of the printer to be deleted.  
                       `<name>` is the name of the printer to be deleted.  
                       `<new_name>` is the new name for this printer.  
                       `<new_server>` is the new print server for this printer.  
                       `<new_address>` is the new IP address for the server.

## **HDR**

This command is identical to the `HEADer` query and is included for backward compatibility purposes.

**Group**     Miscellaneous

**Syntax**     `HDR {OFF|ON|<NR1>}`  
                   `HDR?`

**Related Commands**     [HEADer](#), [VERBose](#)

**Arguments**     `OFF` sets the Response Header Enable State to false. This causes the instrument to omit headers on query responses, so that only the argument is returned.



ON sets the Response Header Enable State to true. This causes the instrument to include headers on applicable query responses. You can then use the query response as a command.

<NR1> = 0 sets the Response Header Enable State to false; any other value sets this state to true, which causes the instrument to omit headers on query responses.

**Examples** HDR OFF specifies that the instrument omits headers on query responses, so that only the argument is returned.

HDR? might return :HEADER 1, indicating that the instrument is including headers on applicable query responses.

## HEADer

This command sets or returns the Response Header Enable State that causes the instrument to either include or omit headers on query responses.

---

**NOTE.** This command does not affect IEEE Std 488.2-1987 Common Commands (those starting with an asterisk); these commands never return headers. This command does make a corresponding change in the Response Header Enable State of the opposite interface (physical or virtual GPIB interface). Refer to Introduction for more information.

---

**Group** Miscellaneous

**Syntax** HEADer {OFF|ON|<NR1>}  
HEADer?

**Related Commands** HDR, VERBose

**Arguments** OFF sets the Response Header Enable State to false. This causes the instrument to omit headers on query responses, so that only the argument is returned.

ON sets the Response Header Enable State to true. This causes the instrument to include headers on applicable query responses. You can then use the query response as a command.

<NR1> = 0 sets the Response Header Enable State to false; any other value sets this state to true, which causes the instrument to omit headers on query responses.

**Examples** HEADER OFF specifies that the instrument omits headers on query responses, so that only the argument is returned.

HEADER? might return :HEADER 1 indicating that the instrument is including headers on applicable query responses.

## HORizontal? (Query Only)

This query-only command returns all settings for the horizontal commands.

<b>Group</b>	Horizontal
<b>Syntax</b>	HORizontal?
<b>Examples</b>	<p>HORIZONTAL? might return the following horizontal settings</p> <pre>:HORIZONTAL:DELAY:MODE 0;POSITION 50.0000;TIME 0.000000000000;HORIZONTAL:MAIN:INTERPRATIO 1.0000;SCALE 40.0000E-9;POSITION 50.0000;SAMPLERATE 1.2500E+9;UNITS:STRING "s";:HORIZONTAL:RECORDLENGTH 500;RESOLUTION 500;ROLL AUTO;FASTFRAME:STATE 1;READOUTS 0;SUMFRAME NONE;MULTIPLEFRAMES:MODE OVERLAY;FRAMESTART:CH1 1;CH2 1;CH3 1;CH4 1;MATH1 1;MATH2 1;MATH3 1;MATH4 1;REF1 1;REF2 1;REF3 1;REF4 1;:HORIZONTAL:FASTFRAME:MULTIPLEFRAMES:NUMFRAMES:CH1 2;CH2 2;CH3 2;CH4 2;MATH1 2;MATH2 2;MATH3 2;MATH4 2;REF1 2;REF2 2;REF3 2;REF4 2;:HORIZONTAL:FASTFRAME:LENGTH 500;COUNT 2;SELECTED:CH1 2;CH2 2;CH3 2;CH4 2;MATH1 2;MATH2 2;MATH3 2;MATH4 2;REF1 2;REF2 2;REF3 2;REF4 2;:HORIZONTAL:FASTFRAME:REF:SOURCE CH4;FRAME 1;:HORIZONTAL:FASTFRAME:TRACK ALL</pre>

## HORizontal:ACQLENGTH? (Query Only)

Returns the record length.

<b>Group</b>	Horizontal
<b>Syntax</b>	HORizontal:ACQLENGTH?

## HORizontal:MAIn? (Query Only)

This query-only command returns the time per division of the time base. This command is equivalent to selecting Position/Scale from the Horiz/Acq menu.

<b>Group</b>	Horizontal
<b>Syntax</b>	<code>HORizontal:MAIn?</code>
<b>Examples</b>	<code>HORIZONTAL:MAIN?</code> might return <code>:HORIZONTAL:MAIN:INTERPRATIO 1.0000;SCALE 40.0000E-9;POSITION 50.0000;SAMPLERATE 1.2500E+9;UNITS:STRING "s"</code>

## **HORizontal[:MAIn]:DELay:MODE**

This command sets or returns the time base trigger delay mode. This command is equivalent to choosing Delay Mode On from the Horiz/Acq menu.

<b>Group</b>	Horizontal
<b>Syntax</b>	<code>HORizontal[:MAIn]:DELay:MODE {ON OFF &lt;NR1&gt;}</code> <code>HORizontal[:MAIn]:DELay:MODE?</code>
<b>Related Commands</b>	<a href="#">HORizontal[:MAIn]:DELay:TIME</a>
<b>Arguments</b>	<p>ON enables the time base trigger delay mode.</p> <p>OFF disables the time base trigger delay mode.</p> <p>&lt;NR1&gt; = 0 disables the time base trigger delay mode, any other value enables the time base trigger delay mode.</p>
<b>Examples</b>	<p><code>HORIZONTAL:DELAY:MODE ON</code> enables the time base trigger delay mode.</p> <p><code>HORIZONTAL:DELAY:MODE?</code> might return <code>:HORIZONTAL:DELAY:MODE 1</code> indicating that the time base trigger delay mode is currently enabled.</p>

## **HORizontal[:MAIn]:DELay:STATE**

This command sets or returns the time base trigger delay mode. This command is equivalent to choosing Delay Mode On from the Horiz/Acq menu.

<b>Group</b>	Horizontal
--------------	------------

**Syntax**      `HORizontal[:MAIN]:DELay:STATE {ON|OFF|<NR1>}`  
`HORizontal[:MAIN]:DELay:STATE?`

**Related Commands**    [HORizontal\[:MAIN\]:DELay:TIME](#)

**Arguments**      ON enables the time base trigger delay mode.  
                      OFF disables the time base trigger delay mode.  
                      <NR1> = 0 disables the time base trigger delay mode, any other value enables the time base trigger delay mode.

**Examples**      `HORIZONTAL:MAIN:DELAY:STATE ON` enables the time base trigger delay mode.  
                      `HORIZONTAL:MAIN:DELAY:STATE?` might return `:HORIZONTAL:DELAY:MODE 1` indicating that the time base trigger delay mode is currently enabled.

## **HORizontal[:MAIN]:DELay:TIME**

This command sets or returns the time base trigger delay time. This command is equivalent to selecting Position/Scale from the Horiz/Acq menu and choosing a value for Horiz Delay.

**Group**          Horizontal

**Syntax**      `HORizontal[:MAIN]:DELay:TIME <NR3>`  
`HORizontal[:MAIN]:DELay:TIME?`

**Related Commands**    [HORizontal\[:MAIN\]:DELay:MODE](#)

**Arguments**      <NR3> specifies the time base trigger delay time setting, typically represented in seconds.

**Examples**      `HORIZONTAL:DELAY:TIME 5.0E-3` sets the time base trigger delay time to 5 ms.  
                      `HORIZONTAL:DELAY:TIME?` might return `:HORIZONTAL:MAIN:DELAY 5.0000E-05` indicating that the time delay setting is 5  $\mu$ s.

## **HORizontal[:MAIN]:POSition? (Query Only)**

This query-only command always returns a horizontal position of 50%.

<b>Group</b>	Horizontal
<b>Syntax</b>	<code>HORizontal[:MAIN]:POSition??</code>
<b>Related Commands</b>	<a href="#">HORizontal[:MAIN]:DELay:MODE</a>
<b>Examples</b>	<code>HORIZONTAL:MAIN:POSITION?</code> returns <code>:HORIZONTAL:MAIN:POSITION 5.0000E+01</code> indicating that the horizontal position of the waveform on the screen is currently set to 50 percent.

## **HORizontal:MAIn:SAMPLERate? (Query Only)**

This query-only command returns the current horizontal sample rate.

<b>Group</b>	Horizontal
<b>Syntax</b>	<code>HORizontal:MAIn:SAMPLERate?</code>
<b>Related Commands</b>	<a href="#">HORizontal:RESOlution</a>
<b>Examples</b>	<code>HORIZONTAL:MAIN:SAMPLERATE?</code> might return <code>:HORIZONTAL:MAIN:SAMPLERATE 2.5000E+09</code> indicating that the sample rate is currently set to 2.5 GS/s.

## **HORizontal[:MAIn]:SCALE**

This command sets the time per division for the time base or returns its horizontal scale on the display and is identical to the `HORizontal:SCALE` command. The specified scale value is rounded to a valid scale setting. This command is equivalent to selecting Position/Scale from the Horiz/Acq menu and then choosing a Scale value.

<b>Group</b>	Horizontal
<b>Syntax</b>	<code>HORizontal[:MAIN]:SCALE</code> <code>HORizontal[:MAIN]:SCALE?</code>
<b>Arguments</b>	<NR3> is the time per division. The range is from 200 ps through 40 s.

**Examples**     `HORIZONTAL[:MAIN]:SCALE 2E-6` sets the main scale to 2µs per division.  
`HORIZONTAL[:MAIN]:SCALE?` might return `:HORIZONTAL:MAIN:SCALE 2.0000E-06` indicating that the main scale is currently set to 2 µs per division.

## HORizontal:MAIn:SECdiv

This command sets the time per division for the time base or returns its horizontal scale on the display and is identical to the [HORizontal\[:MAIn\]:SCAle](#) command. The specified scale value is rounded to a valid scale setting. This command is equivalent to selecting Position/Scale from the Horiz/Acq menu and then choosing a Scale value.

**Group**     Horizontal

**Syntax**     `HORizontal:MAIn:SECdiv`  
`HORizontal:MAIn:SECdiv?`

**Arguments**     <NR3> is the time per division. The range is from 200 ps through 40 s.

**Examples**     `HORIZONTAL:MAIN:SECDIV 2E-6` sets the main scale to 2µs per division.  
`HORIZONTAL:MAIN:SECDIV?` might return `:HORIZONTAL:MAIN:SCALE 2.0000E-06` indicating that the main scale is currently set to 2 µs per division.

## HORizontal:MAIn:UNIts? (Query Only)

This query-only command returns the units for the horizontal time base. It is equivalent to [HORizontal:MAIn:UNIts:STRing?](#).

**Group**     Horizontal

**Syntax**     `HORizontal:MAIn:UNIts?`

**Related Commands**     [HORizontal:MAIn:UNIts:STRing?](#)

**Examples**     `HORIZONTAL:MAIN:UNITS??` might return `:HORIZONTAL:MAIN:UNITS STRING "Hz"`.

## HORizontal:MAIn:UNIts:STRing? (Query Only)

This command sets or returns the units string for the horizontal time base trigger delay.

**Group** Horizontal

**Syntax** HORizontal:MAIn:UNIts:STRing?

**Related Commands** [HORizontal:MAIn:UNIts?](#)

**Examples** HORizontal:MAIn:UNIts:STRing? might return  
:HORizontal:MAIn:UNIts:STRing "Hz" indicating that the  
horizontal units string is set to Hertz.

## HORizontal:PREViewstate? (Query Only)

Returns a boolean value to indicate whether the acquisition system is in the preview state.

**Group** Horizontal

**Syntax** HORizontal:PREViewstate?

**Returns** <NR1> = 1 if the acquisition system is in the preview state.  
<NR1> = 0 if the acquisition system is not in the preview state.

## HORizontal:RECOrdlength

This command sets the horizontal record length to the number of data points in each frame. The query form of this command returns the current horizontal record length. This command is equivalent to selecting Resolution from the Horiz/Acq menu and then entering the desired Rec Length.

**Group** Horizontal

**Syntax** HORizontal:RECOrdlength <NR1>  
HORizontal:RECOrdlength?

**Arguments** <NR1> represents the supported values for horizontal record lengths, which range from 500 through 400K data points. For more information about valid data point ranges, select Specifications from the Help menu and choose the Horizontal & Acquisition tab.

**Examples** HORIZONTAL:RECORDLENGTH 50000 specifies that 50000 data points will be acquired for each record.

HORIZONTAL:RECORDLENGTH? might return :HORIZONTAL:RECORDlength 5000 indicating that the horizontal record length is equal to 5000 data points.

## HORizontal:RESolution

This command sets the horizontal record length to the number of data points in each frame. The sample rate is automatically adjusted at the same time to maintain a constant time per division. The query form of this command returns the current horizontal record length. This command is equivalent to adjusting the RESOLUTION knob on the front panel.

**Group** Horizontal

**Syntax** HORizontal:RESolution <NR1>  
HORizontal:RESolution?

**Arguments** <NR1> represents the supported values for horizontal record lengths, which range from 500 through 400K data points. For more information about valid data point ranges, select Specifications from the Help menu and choose the Horizontal & Acquisition tab.

**Examples** HORIZONTAL:RESOLUTION 50000 specifies that 50000 data points will be acquired for each record.

HORIZONTAL:RESOLUTION? might return :HORIZONTAL:RESOLUTION 5000 indicating that the horizontal record length is equal to 5000 data points.

## HORizontal:ROLL? (Query Only)

This command queries the Roll Mode status. Use Roll Mode when you want to view data at very slow sweep speeds. It is useful for observing data samples on the screen as they occur.

**Group** Horizontal



**Syntax**     `HORizontal:ROLL?`

**Examples**     `HORIZONTAL:ROLL?` might return `:HORIZONTAL:ROLL OFF` indicating that the Roll Mode is disabled.

## ID? (Query Only)

This query-only command returns identifying information about the instrument and related firmware.

**Group**     Miscellaneous

**Syntax**     `ID?`

**Related Commands**     [\\*IDN?](#)

**Examples**     `ID?` might return `:TEK/TDS5404,CF:91.1CT,FV:01.00.912`. This indicates the instrument model number, configured format, and firmware version number.

## \*IDN? (Query Only)

This query-only command returns the instrument identification code.

**Group**     Miscellaneous

**Syntax**     `*IDN?`

**Related Commands**     [ID?](#)

**Examples**     `*IDN?` might return `:TEKTRONIX,TDS5054B,10000001,CF:91.1CT FV:01.00.912`, indicating the instrument model number, serial number, configured number, and firmware version number.

## LANGuage

Sets or returns the user interface display language. This command only affects the oscilloscope displayed language. Remote commands and their responses are always in English.

**Group** Miscellaneous

**Syntax** LANGuage  
{ENGLi sh|FRENch|GERMan|ITALi an|SPANi sh|PORTUGuese|JAPANese|KOREan|RUSSt  
LANGuage?

**Examples** LANGuage? might return :LANGuage ENGLISH.

## LANGuage:INCRe ment (No Query Form)

Changes the oscilloscope display language to the next language in the sequence.

**Group** Miscellaneous

**Syntax** LANGuage:INCRe ment

## LOCK

This command enables or disables all front-panel buttons and knobs, including the touch screen. There is no front panel equivalent.

**Group** Miscellaneous

**Syntax** LOCK {ALL|NONE}  
LOCK?

**Related Commands** [UNLock](#)

**Arguments** ALL disables all front panel controls.  
NONE enables all front panel controls. This is equivalent to the UNLock ALL command.

If the instrument is in the Remote With Lockout State (RWLS), the LOCK NONE command has no effect. For more information, see the ANSI/IEEE Std 488.1-1987 Standard Digital Interface for Programmable Instrumentation, section 2.8.3 on RL State Descriptions.

**Examples** LOCK ALL locks the front panel controls.

LOCK? might return :LOCK NONE indicating that the front panel controls are enabled by this command.

## \*LRN? (Query Only)

This query-only command returns the commands that list the instrument settings (except for configuration information for the calibration values), allowing you to record or "learn" the current instrument settings. You can use these commands to return the instrument to the state it was in when you made the \*LRN? query. This command is identical to the SET? Command.

**Group** Miscellaneous

**Syntax** \*LRN?

**Related Commands** SET?

**Examples** \*LRN? might return the following response: :ACQUIRE:STOPAFTER RUNSTOP;STATE 1;MODE SAMPLE;NUMENV 10;NUMAVG 16;NUMSAMPLES 16000;SAMPLINGMODE IT;:FASTACQ:STATE 0;:APPLICATION:GPKNOB1:ACTIVE 0;:APPLICATION:GPKNOB2:ACTIVE 0;:APPLICATION:WINDOW:HEIGHT 236;WIDTH 640;:APPLICATION:SCOPEAPP:STATE RUNNING;WINDOW FULLSCREEN;:APPLICATION:EXTAPP:STATE NOTRUNNING;:VARIABLE:EVENT:REPORT BOTH;:AUXOUT:SOURCE ATRIGGER;EDGE FALLING;:CMBATCH 1;:HEADER 1;:LOCK NONE;:ROSC:SOURCE INTERNAL;:VERBOSE 1;:ALIAS:STATE 0;:DISPLAY:CLOCK 1;COLOR:PALETTE:IMAGEVIEW TEMPERATURE;RECORDVIEW NORMAL;USER:GRATICULE 165,50,15;CH1 180,50,100;CH2 300,50,100;CH3 60,50,100;CH4 240,50,100;REF1 0,90,0;REF2 0,90,100;REF3 60,90,100;REF4 240,90,100;MATH1 160,50,100;MATH2 40,60,100;MATH3 120,60,100;MATH4 195,50,100;HISTOGRAM 320,50,100;CARET 150,50,100;MASK 0,25,75;MASKHIGHLIGHT 140,50,100;:DISPLAY:COLOR:MATHCOLOR DEFAULT

## MARK (No Query Form)

Move to the next or previous mark on the waveform.

**Group** Mark

**Syntax** MARK {NEXT|PREVIOUS}

## MARK:CREATE (No Query Form)

Creates a mark on a particular waveform or all waveforms in a column.

**Group** Mark

**Syntax** MARK:CREATE {CH<x>|REF<x>|MATH|COLUMN}

**Arguments** CH<x> creates the mark on a channel waveform, where <x> is the channel number.  
REF<x> creates the mark on a reference waveform, where <x> is the reference waveform number.  
MATH creates the mark on the math waveform.  
COLUMN creates marks on all waveforms in the current zoom pixel column.

## MARK:DELEte (No Query Form)

Deletes a mark on a particular waveform, all waveforms in a column, or all marks.

**Group** Mark

**Syntax** MARK:DELEte {CH<x>|REF<x>|MATH|COLUMN|SELECTED|ALL}

**Arguments** CH<x> deletes the marks on a channel waveform, where <x> is the channel number.  
REF<x> deletes the marks on a reference waveform, where <x> is the reference waveform number.  
MATH deletes the marks on the math waveform.  
COLUMN deletes the marks on all waveforms in the current zoom pixel column.

SELECTED deletes the selected mark.

ALL deletes all marks.

## MARK:FREE? (Query Only)

Returns how many marks are free to be used

**Group** Mark

**Syntax** MARK:FREE?

## MARK:SELEcted:END? (Query Only)

Returns the end of the selected mark, in terms of 0 to 100% of the waveform.

**Group** Mark

**Syntax** MARK:SELEcted:END?

## MARK:SELEcted:FOCUS? (Query Only)

Returns the focus of the selected mark, in terms of 0 to 100% of the waveform.

**Group** Mark

**Syntax** MARK:SELEcted:FOCUS?

## MARK:SELEcted:MARKSINCOLumn? (Query Only)

Returns how many marks are in the current zoom pixel column.

**Group** Mark

**Syntax** MARK:SELEcted:MARKSINCOLumn?

## MARK:SElected:OWNer? (Query Only)

Returns the owner of the selected mark, for example, USER, SEARCH1.

**Group** Mark

**Syntax** MARK:SElected:OWNer?

**Returns** <QString> is the owner of the mark.

## MARK:SElected:SOURCE? (Query Only)

Returns the source waveform of the selected mark.

**Group** Mark

**Syntax** MARK:SElected:SOURCE?

## MARK:SElected:START? (Query Only)

Returns the start of the selected mark, in terms of 0 to 100% of the waveform.

**Group** Mark

**Syntax** MARK:SElected:START?

## MARK:SElected:STATe? (Query Only)

Returns the on or off state of the selected mark.

**Group** Mark

**Syntax** MARK:SElected:STATe?

## MARK:SElected:ZOOm:POSition? (Query Only)

Returns the position of the selected mark, in terms of 0 to 100% of the upper window.

**Group** Mark

**Syntax** MARK:SElected:ZOOm:POSition?

## MARK:TOTal? (Query Only)

Returns how many marks are used.

**Group** Mark

**Syntax** MARK:TOTal?

## MATH[1]:DEFIne

This command allows you to define a new waveform using mathematical expressions. The mathematical expression can be a string of the form <wfm><operation><wfm>, where wfm are any combination of live channels or reference waveforms, and operation is any of +, -, \* or /. If the string is in the form FFT(<wfm>) where wfm is any live channel or reference waveform, FFT math is performed. Otherwise, if the contents of the string can be parsed by the advanced math parser without errors, advanced math is performed. The Dual Wfm Math, FFT and Advanced Math menus on the front panel contain controls that allow building equivalent math expressions to those described above.

**Group** Math

**Syntax** MATH[1]:DEFIne <QString>  
MATH[1]:DEFIne?

**Related Commands** [MATHVAR:VAR<x>](#)

**Arguments** <QString> quoted string argument is the mathematical expression that defines the waveform.

**Table 2-35: Advanced Math expression elements**

Expression	Description
CH1-CH4, REF1-REF4	Specifies a waveform data source.
FFT( , INTG( , DIFF(	Executes a Fast Fourier Transform, integration, or differentiation operation on the expression that follows. The FFT operator must be the first (left-most) operator in an expression. All these operations must end with a right parenthesis.
AMPlitude( , AREa( , BURst( , CARea( , CMEan( , CRMs( , DELay( , FALL( , FREQuency( , HIGH( , LOW( , MAXimum( , MEAN( , MINImum( , NDUTy( , NOVershoot( , NWIdth( , PDUTy( , PERIod( , PHAse( , PK2pk( , POVershoot( , PWIdth( , RISe( , RMS(	Executes the selected measurement operation on the waveform (active or reference) that follows. All these operations must end with a right parenthesis.
VAR1, VAR2	Adds the user-defined variable to the expression. Refer to the MATH:VAR<x> command.
+, -, *, /	Executes an addition, subtraction, multiplication, or division operation on the following expression. + and - are also unary; use - to negate the expression that follows.
( ), ,	Parentheses provide a way to control evaluation order in an expression. The comma is used to separate the "from" and "to" waveforms in Delay and Phase measurement operations.
1-0 , . , E	Specifies a numeric value in (optional) scientific notation.

**Examples**    `MATH1:DEFINE " CH1+CH2"` adds the Ch 1 waveform and Ch 2 waveform, storing the results in Math 2.

`MATH:DEFINE?` might return

`:MATH1:DEFINE "CH2*REF2"` as the expression that defines Math 1.

## MATH[1]:HORizontal:POSition

Sets or returns the math horizontal display position for FFT or math waveforms that only have reference waveform source waveforms.

**Group**    Math



<b>Syntax</b>	MATH[1]:HORizontal:POSition <NR3> MATH[1]:HORizontal:POSition?
<b>Arguments</b>	<NR3> is the % of the math waveform (divided by 100) that precedes center screen. It can vary from 0.0e0 to 1.0e0.
<b>Examples</b>	MATH:HORIZONTAL:POSITION 1.0E-1 sets the horizontal position to 10% pretrigger

## MATH[1]:HORizontal:SCALE

Sets or returns the math horizontal display scale for FFT or for dual math waveforms that have source waveforms that are reference waveforms. The horizontal scale of a dual math waveform with a channel source waveform is set through the HORizontal:SCALE command.

<b>Group</b>	Math
<b>Syntax</b>	MATH[1]:HORizontal:SCALE <NR3> MATH[1]:HORizontal:SCALE?
<b>Arguments</b>	<NR3> is the math horizontal scale in seconds.
<b>Examples</b>	MATH:HORIZONTAL:SCALE? might return MATH:HORIZONTAL:SCALE 2.0E-4 indicating that the math horizontal scale is 200 m

## MATH[1]:HORizontal:UNITs

Sets or returns the math waveform horizontal measurement unit value.

<b>Group</b>	Math
<b>Syntax</b>	MATH[1]:HORizontal:UNITs <QString> MATH[1]:HORizontal:UNITs?
<b>Arguments</b>	<QString> is a text label to apply to horizontal units when the horizontal unit is "?" (unknown unit value).

**Examples**     `MATH:HORIZONTAL:UNITS` might return `MATH:HORIZONTAL:UNITS "?"` indicating that the math horizontal unit label for unknown values is the default question mark unit.

## MATH[1]:SPECTral:MAG

This command sets or returns the units of the SpectralMag function in the math string. The Math waveform is specified by *x*, which ranges from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. If you are using the standard math, this command is equivalent to selecting Magnitude Spectrum from the Math menu and then selecting the units that you want from the Scale button drop-down menu. If you are using the Advanced Analysis functions, this command is equivalent to selecting Spectral Setup from the Math menu, choosing the Mag tab, and then clicking the desired Scale button.

**Group**     Math

**Syntax**     `MATH[1]:SPECTral:MAG {LINEAR|DB}`  
`MATH[1]:SPECTral:MAG?`

**Arguments**     `LINEAR` sets the SpectralMag units to linear.  
`DB` sets the SpectralMag units to decibels.

**Examples**     `MATH2:SPECTRAL:MAG DB` sets the SpectralMag units for Math2 to decibels.  
`MATH2:SPECTRAL:MAG?` might return `:MATH2:SPECTRAL:MAG DB` indicating that the SpectralMag units for Math2 are set to decibels.

## MATH[1]:SPECTral:WINDow

This command sets or returns the window function used to multiply the spectral analyzer input data for the specified math waveform. The Math waveform is specified by *x*, which ranges from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. A spectral window determines what the filter shape of the spectral analyzer will be in the frequency domain. It can be described by a mathematical function that is multiplied point-by-point times the input data to the spectral analyzer. This command is equal to selecting Spectral Setup from the Math menu, and choosing from the Window Type drop-down list.

Following is a list of arguments that specify the window function used to multiply the spectral analyzer input data. The windows are listed in the order of their ability to resolve frequencies (resolution bandwidth). For additional information

about spectral windows, see *Selecting a Spectral Window* in the online help for this instrument.

<b>Group</b>	Math
<b>Syntax</b>	MATH[1]:SPECTral:WINDow {RECTangular HAMming HANning BLAckmanharris} MATH[1]:SPECTral:WINDow?
<b>Arguments</b>	RECTangular window function is equivalent to multiplying all gate data by one. HAMming window function is based on a cosine series. HANning window function is based on a cosine series. BLAckmanharris window function is based on a cosine series.
<b>Examples</b>	MATH2:SPECTRAL:WINDOW HANNING applies a Hanning window to the spectral analyzer input data.  MATH2:SPECTRAL:WINDOW? might return :MATH2:SPECTRAL:WINDOW HAMMING indicating that the window function used to multiply the spectral analyzer input data is the Hamming window.

## MATH[1]:TYPe

Sets or returns the math waveform mode type.

<b>Group</b>	Math
<b>Syntax</b>	MATH[1]:TYPe {ADVanced DUAL FFT} MATH[1]:TYPe?
<b>Arguments</b>	ADVanced sets the math waveform mode to advanced math. TDS3AAM only. DUAL sets the math waveform mode to dual waveform math. FFT sets the math waveform mode to FFT math. TDS3AAM, TDS3FFT only.
<b>Examples</b>	MATH:TYPe FFT sets the math waveform mode to FFT.

## MATH[1]:VERTical:POSition

This command sets or returns the vertical position of the specified Math waveform. The position value is usually applied to the signal before it is digitized. The highest three units/div scale ranges of a given math are implemented by changing the way the acquired data is displayed. When the instrument is operating in any of these highest three scale ranges, the position control operates only on the signal after it is digitized. Note that if a signal that exceeds the range of the digitizer in one of these three scale ranges is repositioned, the displayed waveform will contain clipped values on-screen. This command is equivalent to selecting Position/Scale from the Math menu and then entering a Vert Pos value or adjusting the front-panel Vertical POSITION knob.

Increasing the position value of a waveform causes the waveform to move up, and decreasing the position value causes the waveform to move down. Position adjusts only the display position of a waveform, whether a channel, math, or reference waveform. The position value determines the vertical graticule coordinate at which input signal values, equal to the present offset setting for that reference, are displayed. For example, if the position for Math 3 is set to 2.0 and the offset is set to 3.0, then the input signals equal to 3.0 are displayed 2.0 divisions above the center of the screen.

Be aware that autoscaling occurs when a math waveform is first defined and enabled, or when a math string changes. After the math waveform is computed for the first time, the instrument determines the min + max of that waveform data. Then, the instrument sets the math position so that  $(\text{min} + \text{max})/2$  is in the center of the screen. In addition, the instrument sets the math scale so that the range of the min and max cover 6 divisions. This autoscaling process can take up to 1/2 second to complete and will override any vertical scale or position commands for that math waveform received during this time. You should insert an appropriate pause in your program after defining and enabling a math waveform before changing its position or scale.

<b>Group</b>	Math
<b>Syntax</b>	MATH[1]:VERTical:POSITION <NR3> MATH[1]:VERTical:POSITION?
<b>Related Commands</b>	<a href="#">CH&lt;x&gt;:POSition</a> , <a href="#">REF&lt;x&gt;:VERTical:POSition</a>
<b>Arguments</b>	<NR3> is the desired position in divisions from the center graticule.
<b>Examples</b>	MATH2:VERTICAL:POSITION 1.3E+00 positions the Math 2 input signal 1.3 divisions higher than a position of 0.

MATH1:VERTICAL:POSITION? might return :MATH1:VERTICAL:POSITION  
-1.3000E+00 indicating that the current position of Math 1 is 1.3 divisions  
below the center graticule.

## MATH[1]:VERTical:SCAle

This command sets or returns the vertical scale of the specified math waveform. This command is equivalent to selecting Position/Scale from the Math menu and then entering a Vert Scale value or adjusting the front-panel Vertical SCALE knob.

Each waveform has its own vertical scale parameter. For a signal with constant amplitude, increasing the scale causes the waveform to be displayed smaller. Decreasing the scale causes the waveform to be displayed larger.

Scale affects all waveforms. For reference and math waveforms, the scale setting controls the display only, graphically scaling these waveforms and having no affect on the acquisition hardware.

Be aware that autoscaling occurs when a math waveform is first defined and enabled, or when a math string changes. After the math waveform is computed for the first time, the instrument determines the min + max of that waveform data. Then, the instrument sets the math position so that  $(\text{min} + \text{max})/2$  is in the center of the screen. In addition, the instrument sets the math scale so that the range of the min and max covers 6 divisions. This autoscaling process can take up to 1/2 second to complete and will override any vertical scale or position commands for that math waveform received during this time. You should insert an appropriate pause in your program after defining and enabling a math waveform before changing its position or scale.

**Group** Math

**Syntax** MATH[1]:VERTical:SCAle <NR3>  
MATH[1]:VERTical:SCAle?

**Related Commands** CH<x>:SCAle, REF<x>:VERTical:SCAle

**Arguments** <NR3> is the scale, in volts, amps or watts per division. The range is from 100.0E-36 through 100.0E+36.

**Examples** MATH1:VERTICAL:SCALE 100E-03 sets the Math scale to 100 mV per division.  
MATH:VERTICAL:SCALE? might return :MATH:VERTICAL:SCALE  
1.0000E+00 indicating that the current scale setting of Math is 1 V per division.

## MATH[1]:VERTical:UNITs

Sets or returns the math waveform vertical measurement unit value.

<b>Group</b>	Math
<b>Syntax</b>	MATH[1]:VERTical:UNITs <QString> MATH[1]:VERTical:UNITs?
<b>Arguments</b>	<QString> is a text label to apply to vertical units when the vertical unit is "?" (unknown unit value).
<b>Examples</b>	MATH:VERTICAL:UNITs? might return MATH:VERTICAL:UNITs "joules" indicating that the math vertical unit label for unknown values is joules.

## MATHVAR? (Query Only)

This command queries all 8 numerical values you can use within math expressions.

<b>Group</b>	Math
<b>Syntax</b>	MATHVAR?
<b>Related Commands</b>	<a href="#">MATHVAR:VAR&lt;x&gt;</a> , <a href="#">MATH[1]:DEFIne</a>
<b>Returns</b>	<NR3> are the stored numerical values.
<b>Examples</b>	MATHVAR? returns the values of all variables stored in locations 1 through 8.

## MATHVAR:VAR<x>

This command sets or returns one of 8 different numerical values you can use within math expressions. These values can range from -9.9e37 to 9.9e37; the default values are 0.0. <x> specifies the location, 1 through 8, in which you can store values. Stored math variables can be reference within math expressions as VAR1, VAR2,...VAR8.

For example, the following command defines MATH1 as the product of Channel 1 and math variable 1: MATH1:DEFINE "CH1 \* VAR1".

<b>Group</b>	Math
<b>Syntax</b>	MATHVAR:VAR<x> <NR3> MATHVAR:VAR<x>?
<b>Related Commands</b>	<a href="#">MATHVAR:VAR&lt;x&gt;</a> , <a href="#">MATH[1]:DEFIne</a>
<b>Arguments</b>	<NR3> specifies the numerical value to be stored in location x <1 through 8>.
<b>Examples</b>	MATHVAR:VAR3 -2.43E-5 stores the value -2.43e-5 in the third math variable location.  MATHVAR:VAR3? might return :MATHVAR:VAR3 24.3000E-6 for the expression stored in location 3.

## MEASUrement? (Query Only)

This query-only command returns all measurement parameters in the following order: MEAS1, MEAS2, MEAS3, MEAS4, MEAS5, MEAS6, MEAS7, MEAS8, and IMMED.

<b>Group</b>	Measurement
<b>Syntax</b>	MEASUrement?
<b>Examples</b>	MEASUREMENT? might return :MEASUREMENT:GATING OFF;IMMED:TYPE UNDEFINED;UNITS "V";SOURCE1 CH1;SOURCE2 CH1;SOURCE1:SIGTYPE PULSE;:MEASUREMENT:IMMED:SOURCE2:SIGTYPE PULSE;:MEASUREMENT:IMMED:DELAY:EDGE1 RISE;EDGE2 RISE;DIRECTION FORWARDS;:MEASUREMENT:IMMED:REFLEVEL:METHOD PERCENT;ABSOLUTE:HIGH 0.0000;LOW 0.0000;MID1 0.0000;MID2 0.0000;:MEASUREMENT:IMMED:REFLEVEL:PERCENT:HIGH 90.0000;LOW 10.0000;MID1 50.0000;MID2 50.0000;:MEASUREMENT:IMMED:METHOD HISTOGRAM;NOISE HIGH;:MEASUREMENT:MEAS1:STATE 0;TYPE UNDEFINED;UNITS "V";SOURCE1 CH1;SOURCE2 CH1;SOURCE1:SIGTYPE PULSE;:MEASUREMENT:MEAS1:SOURCE2:SIGTYPE PULSE;:MEASUREMENT:MEAS1:DELAY:EDGE1 RISE;EDGE2 RISE;DIRECTION FORWARDS;:MEASUREMENT:MEAS1:REFLEVEL:METHOD

```
PERCENT;ABSOLUTE:HIGH 0.0000;LOW 0.0000;MID1 0.0000;MID2
0.0000;:MEASUREMENT:MEAS1:REFLEVEL:PERCENT:HIGH 90.0000;LOW
10.0000;MID1 50.0000;MID2 50.0000;:MEASUREMENT:MEAS1:METHOD
HISTOGRAM;NOISE HIGH;:MEASUREMENT:MEAS2:STATE
0;TYPE UNDEFINED;UNITS "V";SOURCE1 CH1;SOURCE2
CH1;SOURCE1:SIGTYPE PULSE;:MEASUREMENT:MEAS2:SOURCE2:SIGTYPE
PULSE;:MEASUREMENT:MEAS2:DELAY:EDGE1 RISE;EDGE2
RISE;DIRECTION.
```

## MEASUrement:CLEARSnapshot (No Query Form)

Removes the measurement snapshot display.

**Group** Measurement

**Syntax** MEASUrement:CLEARSnapshot

### Examples

## MEASUrement:GATing

This command specifies or returns the measurement gating setting. This command is equivalent to selecting Gating from the Measure menu and then clicking the desired Measurement Gating setting.

**Group** Measurement

**Syntax** MEASUrement:GATing {OFF|SCREEn|CURSOR}  
MEASUrement:GATing?

**Arguments** OFF turns off measurement gating.  
SCREEn turns on gating, using the left and right edges of the screen.  
CURSOR limits measurements to the portion of the waveform between the vertical bar cursors, even if they are off screen.

**Examples** MEASUREMENT:GATING CURSOR turns on measurement gating using the cursors as limits.  
MEASUREMENT:GATING? might return :MEASUREMENT:GATING CURSOR indicating that measurements are limited to the portion of the waveform between the vertical bar cursors.



## MEASUrement:IMMed? (Query Only)

This query-only command returns all immediate measurement setup parameters.

**Group** Measurement

**Syntax** MEASUrement:IMMed?

**Examples** MEASUREMENT:IMMED? might return :MEASUREMENT:IMMED:TYPE  
AMPLITUDE; UNITS "V";SOURCE1 CH1;SOURCE2 CH1;  
SOURCE1:SIGTYPE PULSE; :MEASUREMENT:IMMED:SOURCE2:SIGTYPE  
PULSE; :MEASUREMENT:IMMED:DELAY:EDGE1 RISE; EDGE2  
RISE;DIRECTION FORWARDS; :MEASUREMENT:IMMED:REFLEVEL:METHOD  
PERCENT; ABSOLUTE:HIGH 0.0000;LOW 0.0000;MID1 0.0000;MID2  
0.0000;:MEASUREMENT:IMMED:REFLEVEL:PERCENT:HIGH 90.0000;LOW  
10.0000;MID1 50.0000;MID2 50.0000;:MEASUREMENT:IMMED:METH  
HISTOGRAM;NOISE HIGH

## MEASUrement:IMMed:DELaY? (Query Only)

This query-only command returns information about the immediate delay measurement. This command is equivalent to selecting Measurement Setup from the Measure menu, choosing the Time tab and then clicking the Delay button.

**Group** Measurement

**Syntax** MEASUrement:IMMed:DELaY?

**Examples** MEASUREMENT:IMMED:DELAY? might return  
:MEASUREMENT:IMMED:DELAY:EDGE1 RISE;EDGE2 RISE;  
DIRECTION FORWARDS

## MEASUrement:IMMed:DELaY:DIREction

This command sets or returns the starting point and direction that determines the delay "to" edge when taking an immediate delay measurement. Use the [MEASUrement:IMMed:SOURCE2](#) command to specify the delay "to" waveform. This command is equivalent to selecting Measurement Setup from the Measure menu, choosing the Time tab, clicking the Delay button to display the delay settings and then clicking the desired Search Direction setting.

<b>Group</b>	Measurement
<b>Syntax</b>	<pre>MEASUREMENT:IMMED:DELAY:DIRECTION {BACKWARDS FORWARDS} MEASUREMENT:IMMED:DELAY:DIRECTION?</pre>
<b>Related Commands</b>	<a href="#">MEASUREMENT:IMMED:SOURCE2</a>
<b>Arguments</b>	<p>BACKWARDS starts the search at the end of the waveform and looks for the last rising or falling edge in the waveform.</p> <p>FORWARDS starts the search at the beginning of the waveform and looks for the first rising or falling edge in the waveform.</p>
<b>Examples</b>	<pre>MEASUREMENT:IMMED:DELAY:DIRECTION FORWARDS</pre> starts searching from the beginning of the waveform record and looks for the first rising or falling edge. <pre>MEASUREMENT:IMMED:DELAY:DIRECTION?</pre> might return <pre>:MEASUREMENT:IMMED:DELAY:DIRECTION BACKWARDS</pre> indicating that searching begins at the end of the waveform record and looks for the last rising or falling edge.

## MEASUREMENT:IMMED:DELAY:EDGE<x>

Sets or returns the slope of the edge that is used for the delay "from" or "to" waveform when taking an immediate delay measurement. The waveform is specified by MEASUREMENT:IMMED:SOURCE<x>.

<b>Group</b>	Measurement
<b>Syntax</b>	<pre>MEASUREMENT:IMMED:DELAY:EDGE&lt;x&gt; {FALL RISe} MEASUREMENT:IMMED:DELAY:EDGE&lt;x&gt;?</pre>
<b>Arguments</b>	<p>&lt;x&gt; specifies which waveform to use, where &lt;x&gt; = 1 is the "from" waveform, and &lt;x&gt; = 2 is the "to" waveform.</p> <p>FALL specifies the falling edge.</p> <p>RISe specifies the rising edge.</p>
<b>Examples</b>	<pre>MEASUREMENT:IMMED:DELAY:EDGE1 RISe</pre> specifies that the "from" waveform rising edge be used for the immediate delay measurement. <pre>MEASUREMENT:IMMED:DELAY:EDGE1?</pre> returns either RISE or FALL.

## MEASUrement:IMMed:SOURCE[1]

This command sets or returns the source for all single channel immediate measurements and specifies the source to measure "from" when taking an immediate delay measurement or phase measurement. This command is equivalent to selecting Measurement Setup from the Measure menu, choosing the Time tab, clicking the Delay button to display the delay settings and then clicking the desired Source1 (From) setting.

---

**NOTE.** *If you do not specify a numerical suffix, the source is assumed to be SOURCE 1.*

---

**Group** Measurement

**Syntax** MEASUrement:IMMed:SOURCE[1] {CH<x>|MATH<y>|REF<x>|HISTogram}  
MEASUrement:IMMed:SOURCE?

**Related Commands** [MEASUrement:IMMed:SOURCE2](#)

**Arguments** CH<x> is an input channel waveform. The x variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.

MATH<y> is a math waveform. The y variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.

REF<x> is a reference waveform. The x variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.

HISTogram indicates histogram as the object to be measured.

**Examples** MEASUREMENT:IMMED:SOURCE1 MATH1  
specifies Math1 as the immediate measurement source.

MEASUREMENT:IMMED:SOURCE? might return  
:MEASUREMENT:IMMED:SOURCE1 CH3 indicating that channel 3  
is the immediate measurement source.

## MEASUrement:IMMed:SOURCE2

This command sets or returns the source to measure "to" for phase or delay immediate measurements. This command is equivalent to selecting Measurement

Setup from the Measure menu, choosing the Time tab, clicking the Delay button to display the delay settings and then clicking the desired Source2 (To) setting.

Tip: Source2 measurements only apply to phase and delay measurement types, which require both a target (Source1) and reference (Source2) source.

<b>Group</b>	Measurement
<b>Syntax</b>	MEASUREMENT:IMMED:SOURCE2 {CH<x> MATH<y> REF<x> HISTogram} MEASUREMENT:IMMED:SOURCE2?
<b>Related Commands</b>	<a href="#">MEASUREMENT:IMMED:SOURCE</a>
<b>Arguments</b>	<p>CH&lt;x&gt; is an input channel waveform. The x variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.</p> <p>MATH&lt;y&gt; is a math waveform. The y variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.</p> <p>REF&lt;x&gt; is a reference waveform. The x variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.</p> <p>HISTogram indicates histogram as the object to be measured.</p>
<b>Examples</b>	<p>MEASUREMENT:IMMED:SOURCE2 REF3 sets the waveform in reference memory location 3 as the delay "to" source when making delay measurements.</p> <p>MEASUREMENT:IMMED:SOURCE2? might return :MEASUREMENT:IMMED:SOURCE2 MATH1 indicating that Math1 is the immediate measurement source.</p>

## MEASUREMENT:IMMED:TYPE

This command sets or returns the immediate measurement type.

<b>Group</b>	Measurement
<b>Syntax</b>	MEASUREMENT:IMMED:TYPE {AMPLitude AREa BURSt CAREa CMEan CRMs DELay DISTDuty EXTINCTDB EXTINCTPCT EXTINCTRATIO EYEHeight EYEWIdth FALL FREQUENCY HIGH HITS LOW MAXimum MEAN MEDIan MINimum NCROSS NDuty NOVershoot NWidth PBASE

PCROSS|PCTCROSS|PDuty|PEAKHits|PERIOD|  
 PHASE|PK2Pk|PKPKJitter|PKPKNoise|  
 POvershoot|PTOT|PWidth|QFACTOR|RISE|  
 RMS|RMSJitter|PMSNoise|SIGMA1|SIGMA2|  
 SIGMA3|SIXSigmaJit|SNRatio|STDdev| UNDEFINED|WAVEFORMS}  
 MEASUREMENT:IMMED:TYPE?

## Arguments

**AMPLitude** measures the amplitude of the selected waveform. In other words, it measures the high value less the low value measured over the entire waveform or gated region.

$$\text{Amplitude} = \text{High} - \text{Low}$$

**AREa** measures the voltage over time. The area is over the entire waveform or gated region and is measured in volt-seconds. The area measured above the ground is positive, while the area below ground is negative.

**BURSt** measures the duration of a burst. The measurement is made over the entire waveform or gated region.

**CAREa** (cycle area) measures the voltage over time. In other words, it measures, in volt-seconds, the area over the first cycle in the waveform or the first cycle in the gated region. The area measured above the common reference point is positive, while the area below the common reference point is negative.

**CMEan** (cycle mean) measures the arithmetic mean over the first cycle in the waveform or the first cycle in the gated region.

**CRMS** (cycle rms) measures the true Root Mean Square voltage over the first cycle in the waveform or the first cycle in the gated region.

**DELAy** measures the time between the middle reference (default = 50%) amplitude point of the source waveform and the destination waveform.

**DISTDuty** (duty cycle distortion) measures the time between the falling edge and the rising edge of the eye pattern at the mid reference level. It is the peak-to-peak time variation of the first eye crossing measured at the mid-reference as a percent of the eye period.

**EXTINCTDB** measures the extinction ratio of an optical waveform (eye diagram). Extinction Ratio (dB) measures the ratio of the average power levels for the logic High to the logic Low of an optical waveform and expresses the result in dB. This measurement only works for fast acquisition signals or a reference waveform saved in fast acquisition mode.

$$\text{Extinction dB} = 10 \times (\log 10 (\text{High} / \text{Low}))$$

**EXTINCTPCT** measures the extinction ratio of the selected optical waveform. Extinction Ratio (%) measures the ratio of the average power levels for the logic Low (off) to the logic (High) (on) of an optical waveform and expresses the result in percent. This measurement only works for fast acquisition signals or a reference waveform saved in fast acquisition mode.

$$\text{Extinction \%} = 100.0 \times (\text{Low} / \text{High})$$

EXTINCRATIO measures the extinction ratio of the selected optical waveform. Extinction Ratio measures the ratio of the average power levels for the logic High to the logic Low of an optical waveform and expresses the result without units. This measurement only works for fast acquisition signals or a reference waveform saved in fast acquisition mode. Extinction ratios greater than 100 or less than 1 generate errors; low must be greater than or equal to 1  $\mu$ W.

$$\text{Extinction Ratio} = (\text{High} / \text{Low})$$

EYEHeight measures the vertical opening of an eye diagram in volts.

EYEWIdth measures the width of an eye diagram in seconds.

FALL measures the time taken for the falling edge of the first pulse in the waveform or gated region to fall from a high reference value (default is 90%) to a low reference value (default is 10%).

FREQuency measures the first cycle in the waveform or gated region. Frequency is the reciprocal of the period and is measured in hertz (Hz), where 1 Hz = 1 cycle per second.

HIGH measures the High reference (100% level, sometimes called Topline) of a waveform.

You can also limit the High measurement (normally taken over the entire waveform record) to a gated region on the waveform.

HITS (histogram hits) measures the number of points in or on the histogram box.

LOW measures the Low reference (0% level, sometimes called Baseline) of a waveform.

MAXimum finds the maximum amplitude. This value is the most positive peak voltage found. It is measured over the entire waveform or gated region. When histogram is selected with the [MEASUrement:METhod](#) command, the maximum measurement measures the voltage of the highest nonzero bin in vertical histograms or the time of the right-most bin in horizontal histograms.

MEAN amplitude measurement finds the arithmetic mean over the entire waveform or gated region. When histogram is selected with the [MEASUrement:METhod](#) command, the mean measurement measures the average of all acquired points within or on the histogram.

MEDian (histogram measurement) measures the middle point of the histogram box. Half of all acquired points within or on the histogram box are less than this value and half are greater than this value.

MINimum finds the minimum amplitude. This value is typically the most negative peak voltage. It is measured over the entire waveform or gated region. When histogram is selected with the [MEASUrement:METhod](#) command, the minimum measurement measures the lowest nonzero bin in vertical histograms or the time of the left-most nonzero bin in the horizontal histograms.

**NCROSS** (timing measurement) measures the time from the trigger point to the first falling edge of the waveform or gated region. The distance (time) is measured at the middle reference amplitude point of the signal.

**NDuty** (negative duty cycle) is the ratio of the negative pulse width to the signal period, expressed as a percentage. The duty cycle is measured on the first cycle in the waveform or gated region.

$$\text{Negative Duty Cycle} = (\text{Negative Width}) / \text{Period} \times 100\%$$

**NOvershoot** (negative overshoot) finds the negative overshoot value over the entire waveform or gated region.

$$\text{Negative Overshoot} = (\text{Low} - \text{Minimum}) / \text{Amplitude} \times 100\%$$

**NWidth** (negative width) measurement is the distance (time) between the middle reference (default = 50%) amplitude points of a negative pulse. The measurement is made on the first pulse in the waveform or gated region.

**PBase** measures the base value used in extinction ratio measurements.

**PCROSS** (timing measurement) measures the time from the trigger point to the first positive edge of the waveform or gated region. The distance (time) is measured at the middle reference amplitude point of the signal.

**PCTCROSS** measures the location of the eye crossing point expressed as a percentage of **EYEHeight**.

$$\text{Crossing percent} = 100 \times [(\text{eye-crossing-point} - \text{PBase}) / (\text{PTOP} - \text{PBase})]$$

**PDuty** (positive duty cycle) is the ratio of the positive pulse width to the signal period, expressed as a percentage. It is measured on the first cycle in the waveform or gated region.

$$\text{Positive Duty Cycle} = (\text{Positive Width}) / \text{Period} \times 100\%$$

**PEAKHits** measures the number of points in the largest bin of the histogram.

**PERIOD** is the time required to complete the first cycle in a waveform or gated region. Period is the reciprocal of frequency and is measured in seconds.

**PHASE** measures the phase difference (amount of time a waveform leads or lags the reference waveform) between two waveforms. The measurement is made between the middle reference points of the two waveforms and is expressed in degrees, where 360° represents one waveform cycle.

**PK2Pk** (peak-to-peak) finds the absolute difference between the maximum and minimum amplitude in the entire waveform or gated region. When histogram is selected with the [MEASurement:METHOD](#) command, the PK2Pk measurement measures the histogram peak to peak difference.

**PKPKJitter** measures the variance (minimum and maximum values) in the time locations of the cross point.

**PKPKNoise** measures the peak-to-peak noise on a waveform at the mid reference level.

**POvershoot**

The positive overshoot amplitude measurement finds the positive overshoot value over the entire waveform or gated region.

$$\text{Positive Overshoot} = (\text{Maximum} - \text{High}) / \text{Amplitude} \times 100\%$$

**PTOT** measures the top value used in extinction ratio measurements.

**PWidth** (positive width) is the distance (time) between the middle reference (default = 50%) amplitude points of a positive pulse. The measurement is made on the first pulse in the waveform or gated region.

**QFactor** measures the quality factor. The Q factor is a figure of merit for an eye diagram, which indicates the vertical eye opening relative to the noise at the low and high logic levels. It is the ratio of the eye size to noise.

**RISe** timing measurement finds the rise time of the waveform. The rise time is the time it takes for the leading edge of the first pulse encountered to rise from a low reference value (default is 10%) to a high reference value (default is 90%).

**RMS** amplitude measurement finds the true Root Mean Square voltage in the entire waveform or gated region.

**RMSJitter** measures the variance in the time locations of the cross point. The RMS jitter is defined as one standard deviation at the cross point.

**RMSNoise** measures the Root Mean Square noise amplitude on a waveform at the mid reference level.

**SIGMA1** (histogram measurement) measures the percentage of points in the histogram that are within one standard deviation of the histogram mean.

**SIGMA2** (histogram measurement) measures the percentage of points in the histogram that are within two standard deviations of the histogram mean.

**SIGMA3** (histogram measurement) measures the percentage of points in the histogram that are within three standard deviations of the histogram mean.

**SIXSigmajit** (histogram measurement) is  $\text{six} \times \text{RMSJitter}$ .

**SNRatio** measures the signal-to-noise ratio. The signal-to-noise ratio is the amplitude of a noise rejection band centered on the mid level.

**STDdev** measures the standard deviation (Root Mean Square (RMS) deviation) of all acquired points within or on the histogram box.

**UNDEFINED** is the default measurement type, which indicates that no measurement type is specified. Once a measurement type is chosen, it can be cleared using this argument.



WAVEFORMS (waveform count) measures the number of waveforms used to calculate the histogram.

- Examples**     `MEASUREMENT:IMMED:TYPE FREQUENCY` defines the immediate measurement to be a frequency measurement.
- `MEASUREMENT:IMMED:TYPE?` might return `:MEASUREMENT:IMMED:TYPE RMS` indicating that the immediate measurement is the true Root Mean Square voltage.

## MEASUREMENT:IMMED:UNITS? (Query Only)

This query-only command returns the units of the immediate measurement.

**Group**     Measurement

**Syntax**     `MEASUREMENT:IMMED:UNITS?`

- Examples**     `MEASUREMENT:IMMED:UNITS?` might return
- `:MEASUREMENT:IMMED:UNITS "s"`
- indicating that units for the immediate measurement are in seconds.

## MEASUREMENT:IMMED:VALUE? (Query Only)

This query-only command returns the value of the measurement specified by the [MEASUREMENT:IMMED:TYPE](#) command. The measurement is immediately taken on the source(s) specified by a [MEASUREMENT:IMMED:SOURCE](#) command.

---

**NOTE.** *A change to `HORIZONTAL:MAIN:SCALE` or `CH<x>:SCALE` will not necessarily have taken affect if followed by this command.*

---

**Group**     Measurement

**Syntax**     `MEASUREMENT:IMMED:VALUE?`

**Related Commands**     [MEASUREMENT:IMMED:TYPE](#), [MEASUREMENT:IMMED:SOURCE](#), [\\*ESR?](#), [ALLEV?](#)

**Examples** MEASUREMENT:IMMED:VALUE? might return :MEASUREMENT:IMMED:VALUE 9.9000E+37. If the measurement has an error or warning associated with it, then an item is added to the error queue. The error can be checked for with the \*ESR? and ALLEv? commands.

## MEASUREMENT:INDICATORS? (Query Only)

Returns all measurement indicator parameters.

**Group** Measurement

**Syntax** MEASUREMENT:INDICATORS?

**Examples** MEASUREMENT:INDICATORS? might return  
 MEASUREMENT:INDICATORS:STATE MEAS1;NUMHORZ  
 0;NUMVERT 4;HORZ1 7.5E0;HORZ2 -3.400000095367E0;HORZ3  
 0.0E0;HORZ4 0.0E0;VERT1 -6.351123E-6;VERT2  
 -3.179753E-6;VERT3 -6.40943E-6;VERT4 -6.403E-6

## MEASUREMENT:INDICATORS:HORIZ<x>? (Query Only)

Returns the position of the specified horizontal measurement indicator <x>, where <x> can be 1, 2, 3, or 4.

**Group** Measurement

**Syntax** MEASUREMENT:INDICATORS:HORIZ<x>?

**Examples** MEASUREMENT:INDICATORS:HORIZ1? might return  
 MEASUREMENT:INDICATORS:HORIZ1 -2.0E-3 indicating that horizontal  
 indicator1 has a value of -2mV.

## MEASUREMENT:INDICATORS:NUMHORZ? (Query Only)

Returns the number of horizontal measurement indicators currently being displayed.

**Group** Measurement

**Syntax**    `MEASurement:INDICators:NUMHORZ?`

**Examples**    `MEASUREMENT:INDICATORS:NUMHORZ?` might return `MEASUREMENT:INDICATORS:NUMHORZ 2` indicating that there are currently 2 horizontal lines drawn on the graticule showing where the measurement specified by `MEASurement:INDICators:STATE` is being performed.

## **MEASurement:INDICators:NUMVERT? (Query Only)**

Returns the number of vertical measurement indicators currently being displayed.

**Group**    Measurement

**Syntax**    `MEASurement:INDICators:NUMVERT?`

**Examples**    `MEASUREMENT:INDICATORS:NUMVERT?` might return `MEASUREMENT:INDICATORS:NUMVERT 2` indicating that there are currently 2 vertical lines drawn on the graticule showing where the measurement specified by `MEASurement:INDICators:STATE` is being performed.

## **MEASurement:INDICators:STATE**

Set or returns the state of visible measurement indicators. Performs the same operations as the front panel Measurement Indicators menu.

**Group**    Measurement

**Syntax**    `MEASurement:INDICators:STATE {OFF|MEAS<x>}`  
`MEASurement:INDICators:STATE?`

**Arguments**    `OFF` turns off visible measurement indicators.  
`MEAS<x>` turns on the display of visible measurement indicators for measurement `<x>`, where `<x>` can be 1, 2, 3, or 4. There must be an active measurement before you can activate an indicator for a specified measurement.

**Examples**    `MEASUREMENT:INDICATORS:STATE MEAS2` turns on the display of visible measurement indicators for measurement 2.

MEASUREMENT:INDICATORS:STATE? might return  
 MEASUREMENT:INDICATORS:STATE OFF indicating that no  
 measurement indicators are active.

## MEASUrement:INDICators:VERT<x>? (Query Only)

Returns the value of the specified vertical measurement indicator <x> from the trigger point, where <x> can be 1, 2, 3, or 4. A negative value means that the indicator is positioned earlier in the waveform record than the trigger point.

**Group** Measurement

**Syntax** MEASUrement:INDICators:VERT<x>?

**Examples** MEASUREMENT:INDICATORS:VERT2? might  
 return MEASUREMENT:INDICATORS:VERT2 -3.724507E-6  
 indicating that the second measurement indicator is positioned 3.72 ms before  
 the trigger point.

## MEASUrement:MEAS<x>? (Query Only)

This query-only command returns all measurement parameters for the displayed measurement specified by x, which can range from 1 through 8. This query command is equivalent to selecting Measurement Setup from the Measure menu and viewing the Measurements table; then choosing the Time tab, clicking the Delay button and viewing the Delay Edge and Search Direction settings.

**Group** Measurement

**Syntax** MEASUrement:MEAS<x>?

**Examples** MEASUREMENT:MEAS<x>? might return :MEASUREMENT:MEAS1:STATE  
 0;TYPE UNDEFINED;UNITS "V";SOURCE1 CH1;SOURCE2  
 CH1;SOURCE1:SIGTYPE PULSE;:MEASUREMENT:MEAS1:SOURCE2:SIGTYPE  
 PULSE;:MEASUREMENT:MEAS1:DELAY:EDGE1 RISE;EDGE2 RISE;  
 DIRECTION FORWARDS;:MEASUREMENT:MEAS1:REFLEVEL:METHOD  
 PERCENT;ABSOLUTE:HIGH 0.0000;LOW 0.0000;MID1 0.0000;MID2  
 0.0000;:MEASUREMENT:MEAS1:REFLEVEL:PERCENT:HIGH 90.0000;LOW  
 10.0000;MID1 50.0000;MID2 50.0000;:MEASUREMENT:MEAS1:METH  
 ODHISTOGRAM;NOISE HIGH.

## MEASUrement:MEAS<x>:COUNT? (Query Only)

This query-only command returns the number of values accumulated for this measurement since the last statistical reset. Some values may have been ignored because they generated an error. Measurements are specified by x, which ranges from 1 through 8.

**Group** Measurement

**Syntax** MEASUrement:MEAS<x>:COUNT?

**Examples** MEASUREMENT:MEAS3:COUNT? might return :MEASUREMENT:MEAS3:COUNT 3.247000000E+03.

## MEASUrement:MEAS<x>:DELay? (Query Only)

This query-only command returns the delay measurement parameters for the measurement specified by <x>, which ranges from 1 through 8. Measurement parameters are presented in the following order: Edge1, Edge2, and Direction. This query command is equivalent to selecting Time from the Measure menu, choosing Delay and viewing the Delay Edge and Search Direction settings.

**Group** Measurement

**Syntax** MEASUrement:MEAS<x>:DELay?

**Examples** MEASUREMENT:MEAS3? might return :MEASUREMENT:MEAS3:DELAY:EDGE1 RISE;EDGE2 RISE;DIRECTION FORWARDS.

## MEASUrement:MEAS<x>:DELay:DIREction

This command sets or returns the starting point and direction that determines the delay "to" edge when taking a delay measurement. Use the [MEASUrement:MEAS<x>:SOURCE2](#) command to specify the waveform. This command is equivalent to selecting Time from the Measure menu, choosing Delay from the drop-down list and then clicking the desired Search Direction setting. Measurements are specified by x, which ranges from 1 through 8.

**Group** Measurement

<b>Syntax</b>	<pre>MEASUREMENT:MEAS&lt;x&gt;:DELAY:DIRECTION {BACKWARDS FORWARDS} MEASUREMENT:MEAS&lt;x&gt;:DELAY:DIRECTION?</pre>
<b>Related Commands</b>	<a href="#">MEASUREMENT:MEAS&lt;x&gt;:SOURCE2</a>
<b>Arguments</b>	<p>BACKWARDS means that the search starts at the end of the waveform and looks for the last rising or falling edge in the waveform. Use the <a href="#">MEASUREMENT:MEAS&lt;x&gt;:DELAY:EDGE&lt;x&gt;</a> command to specify the slope of the edge.</p> <p>FORWARDS means that the search starts at the beginning of the waveform and looks for the first rising or falling edge in the waveform. Use the <a href="#">MEASUREMENT:MEAS&lt;x&gt;:DELAY:EDGE&lt;x&gt;</a> command to specify the slope of the edge.</p>
<b>Examples</b>	<p>MEASUREMENT:MEAS&lt;x&gt;:DELAY:DIRECTION BACKWARDS starts searching from the end of the waveform record.</p> <p>MEASUREMENT:MEAS&lt;x&gt;:DELAY:DIRECTION? might return :MEASUREMENT:MEAS3:DELAY:DIRECTION BACKWARDS indicating that the current search direction is backwards.</p>

## MEASUREMENT:MEAS<x>:DELAY:EDGE<x>

Sets or returns the slope of the edge that is used for the delay "from" or "to" waveform when taking an immediate delay measurement. The waveform is specified by [MEASUREMENT:MEAS<x>:SOURCE\[1\]](#).

<b>Group</b>	Measurement
<b>Syntax</b>	<pre>MEASUREMENT:MEAS&lt;x&gt;:DELAY:EDGE&lt;x&gt; {FALL RISe} MEASUREMENT:MEAS&lt;x&gt;:DELAY:EDGE&lt;x&gt;?</pre>
<b>Arguments</b>	<p>&lt;x&gt; specifies which waveform to use, where &lt;x&gt; = 1 is the "from" waveform, and &lt;x&gt; = 2 is the "to" waveform.</p> <p>FALL specifies the falling edge.</p> <p>RISe specifies the rising edge.</p>
<b>Examples</b>	<p>MEASUREMENT:MEAS1:DELAY:EDGE1 RISe specifies that the "from" waveform rising edge be used for the immediate delay measurement.</p> <p>MEASUREMENT:MEAS1:DELAY:EDGE1? returns either RISE or FALL.</p>

## MEASUrement:MEAS<x>:MAXimum? (Query Only)

This query-only command returns the maximum value found for this measurement since the last statistical reset. Measurements are specified by x, which ranges from 1 through 8.

**Group** Measurement

**Syntax** MEASUrement:MEAS<x>:MAXimum?

**Examples** MEASUREMENT:MEAS3:MAXIMUM? might return  
:MEASUREMENT:MEAS3:MAXIMUM 4.18E-9.

## MEASUrement:MEAS<x>:MEAN? (Query Only)

This query-only command returns the mean value accumulated for this measurement since the last statistical reset. Measurements are specified by x, which ranges from 1 through 8.

**Group** Measurement

**Syntax** MEASUrement:MEAS<x>:MEAN?

**Examples** MEASUREMENT:MEAS1:MEAN? might return :MEASUREMENT:MEAS1:MEAN  
514.71E-09.

## MEASUrement:MEAS<x>:MINimum? (Query Only)

This query-only command returns the minimum value found for this measurement since the last statistical reset. Measurements are specified by x, which ranges from 1 through 8.

**Group** Measurement

**Syntax** MEASUrement:MEAS<x>:MINimum?

**Examples** MEASUREMENT:MEAS1:MINIMUM? might return  
:MEASUREMENT:MEAS1:MINIMUM 1.75E-09.

## MEASUrement:MEAS<x>:SOURCE[1]

This command sets or returns the source for all single channel measurements and specifies the source to measure "from" when taking a delay measurement or phase measurement. Measurements are specified by x, which ranges from 1 through 8. This command is equivalent to selecting Measurement Setup from the Measure menu and then choosing the desired measurement source.

**Group** Measurement

**Syntax** MEASUrement:MEAS<x>:SOURCE[1]  
{CH<x>|MATH<y>|REF<x>|HISTogram}  
MEASUrement:MEAS<x>:SOURCE[1]?

**Arguments** CH<x> is an input channel waveform. The x variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.

MATH<y> is a math waveform. The y variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.

REF<x> is a reference waveform. The x variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.

HISTogram is a histogram.

**Examples** MEASUREMENT:MEAS2:SOURCE1 MATH1 specifies Math 1 as the measurement 2 source.

MEASUREMENT:MEAS1:SOURCE1? might return  
:MEASUREMENT:MEAS1:SOURCE[1] MATH1 indicating that  
Math1 is the measurement 2 source.

## MEASUrement:MEAS<x>:SOURCE2

This command sets or returns the source for all single channel measurements and specifies the reference source to measure "to" when taking a delay measurement or phase measurement. Measurements are specified by x, which ranges from 1 through 8. This command is equivalent to selecting Measurement Setup from the Measure menu, selecting a measurement type of either Phase or Delay, and then choosing the desired measurement source.

Tip: Source2 measurements only apply to phase and delay measurement types, which require both a target (Source1) and reference (Source2) source.



<b>Group</b>	Measurement
<b>Syntax</b>	MEASUrement:MEAS<x>:SOURCE2 {CH<x> MATH<y> REF<x>} MEASUrement:MEAS<x>:SOURCE2?
<b>Related Commands</b>	<a href="#">MEASUrement:MEAS&lt;x&gt;:TYPE</a>
<b>Arguments</b>	<p>CH&lt;x&gt; is an input channel waveform. The x variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.</p> <p>MATH&lt;y&gt; is a math waveform. The y variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.</p> <p>REF&lt;x&gt; is a reference waveform. The x variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.</p>
<b>Examples</b>	<p>MEASUREMENT:MEAS4:SOURCE2 CH1 specifies CH1 as the delay "to" source when making delay measurement.</p> <p>MEASUREMENT:MEAS2:SOURCE2? might return :MEASUREMENT:MEAS2:SOURCE2 MATH1 indicating that Math 1 is the measurement 2 source.</p>

## MEASUrement:MEAS<x>:STATE

This command sets or returns whether the specified measurement slot is computed and displayed. The measurement slot is specified by x, which ranges from 1 through 8. This command is equivalent to selecting Measurement Setup from the Measure menu and then clicking the Display button.

For a measurement to display, you must have selected a source waveform and defined the measurement you want to take and display. You select the measurement using the [MEASUrement:MEAS<x>:SOURCE\[1\]](#) command. You define the measurement type using the [MEASUrement:MEAS<x>:TYPE](#) command.

<b>Group</b>	Measurement
<b>Syntax</b>	MEASUrement:MEAS<x>:STATE {OFF ON <NR1>} MEASUrement:MEAS<x>:STATE?
<b>Related Commands</b>	<a href="#">MEASUrement:MEAS&lt;x&gt;:SOURCE[1]</a> , <a href="#">MEASUrement:MEAS&lt;x&gt;:TYPE</a>

<b>Arguments</b>	OFF disables calculation and display of the specified measurement slot.
	ON enables calculation and display of the specified measurement slot.
	<NR1> = 0 disables calculation and display of the specified measurement slot; any other value enables calculation and display of the specified measurement slot.
<b>Examples</b>	MEASUREMENT:MEAS2:STATE ON computes and displays the measurement defined as measurement 2.
	MEASUREMENT:MEAS1:STATE? might return :MEASUREMENT:MEAS1:STATE 0 indicating that measurement defined for measurement slot 1 is disabled.

## MEASUrement:MEAS<x>:STDdev? (Query Only)

This query-only command returns the standard deviation of values accumulated for this measurement since the last statistical reset. Measurements are specified by x, which ranges from 1 through 8.

<b>Group</b>	Measurement
<b>Syntax</b>	MEASUrement:MEAS<x>:STDdev?
<b>Examples</b>	MEASUREMENT:MEAS1:STDDEV? might return :MEASUREMENT:MEAS1:STDDEV 21.0E-12.

## MEASUrement:MEAS<x>:TYPE

This command sets or returns the measurement type defined for the specified measurement slot. The measurement slot is specified by x, which ranges from 1 through 8. This command is equivalent to selecting Measurement Setup from the Measure menu and then choosing the desired measurement type.

<b>Group</b>	Measurement
<b>Syntax</b>	MEASUrement:MEAS<x>:TYPE {AMPliitude AREa BURSt CAREa CMEan CRMS DELay DISTDuty EXTINCTDB EXTINCTPCT EXTINCTRATIO EYEHeight EYEWIdth FALL FREQuency HIGH HITS LOW MAXimum MEAN MEDIan MINImum NCROSS NDuty NOvershoot NWIdth PBASE PCROSS PCTCROSS PDuty PEAKHits PERIOD PHase PK2Pk PKPKJitter PKPKNoise POvershoot PTOT PWidth QFACTOR

```
RISe|RMS|RMSJitter|PMSNoise|SIGMA1|SIGMA2|
SIGMA3|SIXSigmajit|SNRatio|STDdev|UNDEFINED| WAVEFORMS}
MEASurement:MEAS<x>:TYPE?
```

## Arguments

**AMPli tude** measures the amplitude of the selected waveform. In other words, it measures the high value less the low value measured over the entire waveform or gated region.

$$\text{Amplitude} = \text{High} - \text{Low}$$

**AREa** measures the voltage over time. The area is over the entire waveform or gated region and is measured in volt-seconds. The area measured above the ground is positive, while the area below ground is negative.

**BURSt** measures the duration of a burst. The measurement is made over the entire waveform or gated region.

**CAREa** (cycle area) measures the voltage over time. In other words, it measures, in volt-seconds, the area over the first cycle in the waveform or the first cycle in the gated region. The area measured above the common reference point is positive, while the area below the common reference point is negative.

**CMEan** (cycle mean) measures the arithmetic mean over the first cycle in the waveform or the first cycle in the gated region.

**CRMS** (cycle rms) measures the true Root Mean Square voltage over the first cycle in the waveform or the first cycle in the gated region.

**DELay** measures the time between the middle reference (default = 50%) amplitude point of the source waveform and the destination waveform.

**DISTDuty** (duty cycle distortion) measures the time between the falling edge and the rising edge of the eye pattern at the mid reference level. It is the peak-to-peak time variation of the first eye crossing measured at the mid-reference as a percent of the eye period.

**EXTINCTDB** measures the extinction ratio of an optical waveform (eye diagram). Extinction Ratio (dB) measures the ratio of the average power levels for the logic High to the logic Low of an optical waveform and expresses the result in dB. This measurement only works for fast acquisition signals or a reference waveform saved in fast acquisition mode.

$$\text{Extinction dB} = 10 \times (\log 10 (\text{High} / \text{Low}))$$

**EXTINCTPCT** measures the extinction ratio of the selected optical waveform. Extinction Ratio (%) measures the ratio of the average power levels for the logic Low (off) to the logic (High) (on) of an optical waveform and expresses the result in percent. This measurement only works for fast acquisition signals or a reference waveform saved in fast acquisition mode.

$$\text{Extinction \%} = 100.0 \times (\text{Low} / \text{High})$$

**EXTINCTRATIO** measures the extinction ratio of the selected optical waveform. Extinction Ratio measures the ratio of the average power levels for the logic High to the logic Low of an optical waveform and expresses the result without units. This measurement only works for fast acquisition signals or a reference waveform saved in fast acquisition mode. Extinction ratios greater than 100 or less than 1 generate errors; low must be greater than or equal to 1  $\mu$ W.

$$\text{Extinction Ratio} = (\text{High} / \text{Low})$$

**EYEHeight** measures the vertical opening of an eye diagram in volts.

**EYEWidth** measures the width of an eye diagram in seconds.

**FALL** measures the time taken for the falling edge of the first pulse in the waveform or gated region to fall from a high reference value (default is 90%) to a low reference value (default is 10%).

**FREQuency** measures the first cycle in the waveform or gated region. Frequency is the reciprocal of the period and is measured in hertz (Hz), where 1 Hz = 1 cycle per second.

**HIGH** measures the High reference (100% level, sometimes called Topline) of a waveform.

You can also limit the High measurement (normally taken over the entire waveform record) to a gated region on the waveform.

**HITS** (histogram hits) measures the number of points in or on the histogram box.

**LOW** measures the Low reference (0% level, sometimes called Baseline) of a waveform.

**MAXimum** finds the maximum amplitude. This value is the most positive peak voltage found. It is measured over the entire waveform or gated region. When histogram is selected with the **MEASUrement:METHod** command, the maximum measurement measures the voltage of the highest nonzero bin in vertical histograms or the time of the right-most bin in horizontal histograms.

**MEAN** amplitude measurement finds the arithmetic mean over the entire waveform or gated region. When histogram is selected with the **MEASUrement:METHod** command, the mean measurement measures the average of all acquired points within or on the histogram.

**MEDian** (histogram measurement) measures the middle point of the histogram box. Half of all acquired points within or on the histogram box are less than this value and half are greater than this value.

**MINIMUM** finds the minimum amplitude. This value is typically the most negative peak voltage. It is measured over the entire waveform or gated region. When histogram is selected with the **MEASUrement:METHod** command, the minimum measurement measures the lowest nonzero bin in vertical histograms or the time of the left-most nonzero bin in the horizontal histograms.

**NCROSS** (timing measurement) measures the time from the trigger point to the first falling edge of the waveform or gated region. The distance (time) is measured at the middle reference amplitude point of the signal.

**NDuty** (negative duty cycle) is the ratio of the negative pulse width to the signal period, expressed as a percentage. The duty cycle is measured on the first cycle in the waveform or gated region.

$$\text{Negative Duty Cycle} = (\text{Negative Width}) / \text{Period} \times 100\%$$

**NOvershoot** (negative overshoot) finds the negative overshoot value over the entire waveform or gated region.

$$\text{Negative Overshoot} = (\text{Low} - \text{Minimum}) / \text{Amplitude} \times 100\%$$

**NWidth** (negative width) measurement is the distance (time) between the middle reference (default = 50%) amplitude points of a negative pulse. The measurement is made on the first pulse in the waveform or gated region.

**PBase** measures the base value used in extinction ratio measurements.

**PCROSS** (timing measurement) measures the time from the trigger point to the first positive edge of the waveform or gated region. The distance (time) is measured at the middle reference amplitude point of the signal.

**PCTCROSS** measures the location of the eye crossing point expressed as a percentage of **EYEHeight**.

$$\text{Crossing percent} = 100 \times [(\text{eye-crossing-point} - \text{PBase}) / (\text{PTOP} - \text{PBase})]$$

**PDuty** (positive duty cycle) is the ratio of the positive pulse width to the signal period, expressed as a percentage. It is measured on the first cycle in the waveform or gated region.

$$\text{Positive Duty Cycle} = (\text{Positive Width}) / \text{Period} \times 100\%$$

**PEAKHits** measures the number of points in the largest bin of the histogram.

**PERIOD** is the time required to complete the first cycle in a waveform or gated region. Period is the reciprocal of frequency and is measured in seconds.

**PHASE** measures the phase difference (amount of time a waveform leads or lags the reference waveform) between two waveforms. The measurement is made between the middle reference points of the two waveforms and is expressed in degrees, where 360° represents one waveform cycle.

**PK2Pk** (peak-to-peak) finds the absolute difference between the maximum and minimum amplitude in the entire waveform or gated region. When histogram is selected with the [MEASUREMENT:METHod](#) command, the PK2Pk measurement measures the histogram peak to peak difference.

**PKPKJitter** measures the variance (minimum and maximum values) in the time locations of the cross point.

**PKPKNoise** measures the peak-to-peak noise on a waveform at the mid reference level.

**POvershoot**

The positive overshoot amplitude measurement finds the positive overshoot value over the entire waveform or gated region.

$$\text{Positive Overshoot} = (\text{Maximum} - \text{High}) / \text{Amplitude} \times 100\%$$

**PTOT** measures the top value used in extinction ratio measurements.

**PWidth** (positive width) is the distance (time) between the middle reference (default = 50%) amplitude points of a positive pulse. The measurement is made on the first pulse in the waveform or gated region.

**QFactor** measures the quality factor. The Q factor is a figure of merit for an eye diagram, which indicates the vertical eye opening relative to the noise at the low and high logic levels. It is the ratio of the eye size to noise.

**RISe** timing measurement finds the rise time of the waveform. The rise time is the time it takes for the leading edge of the first pulse encountered to rise from a low reference value (default is 10%) to a high reference value (default is 90%).

**RMS** amplitude measurement finds the true Root Mean Square voltage in the entire waveform or gated region.

**RMSJitter** measures the variance in the time locations of the cross point. The RMS jitter is defined as one standard deviation at the cross point.

**RMSNoise** measures the Root Mean Square noise amplitude on a waveform at the mid reference level.

**SIGMA1** (histogram measurement) measures the percentage of points in the histogram that are within one standard deviation of the histogram mean.

**SIGMA2** (histogram measurement) measures the percentage of points in the histogram that are within two standard deviations of the histogram mean.

**SIGMA3** (histogram measurement) measures the percentage of points in the histogram that are within three standard deviations of the histogram mean.

**SIXSigmajit** (histogram measurement) is  $\text{six} \times \text{RMSJitter}$ .

**SNRatio** measures the signal-to-noise ratio. The signal-to-noise ratio is the amplitude of a noise rejection band centered on the mid level.

**STDdev** measures the standard deviation (Root Mean Square (RMS) deviation) of all acquired points within or on the histogram box.

**UNDEFINED** is the default measurement type, which indicates that no measurement type is specified. Once a measurement type is chosen, it can be cleared using this argument.

WAVEFORMS (waveform count) measures the number of waveforms used to calculate the histogram.

**Examples**      `MEASUREMENT:MEAS2:TYPE FREQUENCY` defines measurement 2 as a measurement of the frequency of a waveform.

`MEASUREMENT:MEAS1:TYPE?` might return `:MEASUREMENT:MEAS1:TYPE RMS` indicating that measurement 1 is defined to measure the RMS value of a waveform.

## MEASUrement:MEAS<x>:UNIts? (Query Only)

This query-only command returns the units associated with the specified measurement. Measurements are specified by x, which ranges from 1 through 8.

**Group**      Measurement

**Syntax**      `MEASUrement:MEAS<x>:UNIts?`

**Related Commands**      [MEASUrement:MEAS<x>:TYPe](#)

**Examples**      `MEASUREMENT:MEAS1:UNITS?` might return `:MEASUREMENT:MEAS1:UNIts %` indicating units for measurement 1 are set to percent.

## MEASUrement:MEAS<x>:VALue? (Query Only)

This query-only command returns the value that has been calculated for the measurement specified by <x>, which ranges from 1 through 8. This command is equivalent to selecting Display Statistics from the Measure menu and then choosing Value from the drop-down list to display *all* measurement values on-screen.

---

**NOTE.** *This is the displayed value in the on-screen display. If measurement statistics are enabled, a new value is calculated with every waveform. In addition, this value is updated about every 1/3 second. If you are acquiring a long acquisition record, the instrument may take longer to update.*

---

**Group**      Measurement

**Syntax**      `MEASUrement:MEAS<x>:VALue?`

**Related Commands**     [MEASUrement:MEAS<x>:UNIts?](#), [\\*ESR?](#), [ALLEv?](#)

**Examples**     MEASUREMENT:MEAS1:VALUE? might return :MEASUREMENT:MEAS1:VALUE 2.8740E-06. If the measurement has an error or warning associated with it, then an item is added to the error queue. The error can be checked for with the [\\*ESR?](#) and [ALLEv?](#) commands.

## MEASUrement:METHod

This command sets or returns the method used to calculate the 0% and 100% reference level. This command is equivalent to selecting Reference Levels from the Measure menu and then choosing the desired Determine Base, Top From setting.

**Group**     Measurement

**Syntax**     MEASUrement:METHod {Auto|HISTogram|MINMax}  
MEASUrement:METHod?

**Related Commands**     [MEASUrement:REFLevel:PERCent:HIGH](#), [MEASUrement:REFLevel:PERCent:LOW](#), [MEASUrement:REFLevel:PERCent:MID](#), [MEASUrement:REFLevel:PERCent:MID2](#)

**Arguments**     HISTogram sets the high and low reference levels to the most common values either above or below the mid point, depending on whether the high reference point or the low reference point is being defined. Because the statistical approach ignores short-term aberrations, such as overshoot or ringing, the histogram method is the best setting for examining pulses.

MINMax uses the highest and lowest values of the waveform record. This selection is best for examining waveforms with no large, flat portions of a common value, such as sine waves and triangle waves.

Auto chooses the best method (histogram or minmax) for each dataset.

**Examples**     MEASUREMENT:METHod HISTOGRAM specifies that the high and low reference levels are set statistically.

MEASUREMENT:METHod? might return :MEASUREMENT:METHod MINMAX indicating that the reference levels are set to MIN and MAX.



## MEASUrement:REFLevel? (Query Only)

This query-only command returns the current reference level parameters. This command is equivalent to selecting Reference Levels from the Measure menu, and then viewing the current Reference Levels settings.

**Group** Measurement

**Syntax** MEASUrement:REFLevel?

**Examples** MEASUREMENT:REFLEVEL? might return these reference level settings  
 :MEASUREMENT:REFLEVEL:METHOd ABSOLUTE; ABSOLUTE:HIGH  
 0.0000E+00;LOW 0.0000E+00; MID1 0.0000E+00;MID2 0.0000E+00;  
 :MEASUREMENT:REFLEVEL:PERCENT:HIGH 9.0000E+01;LOW  
 1.0000+01;MID1 5.0000E+01; MID2 5.0000E+01

## MEASUrement:REFLevel:ABSolute:HIGH

This command sets or returns the high reference level, and is the upper reference level when [MEASUrement:REFLevel:METHOd](#) is set to Absolute. This command affects the results of rise and fall measurements and is equivalent to selecting Reference Levels from the Measure menu, and then entering the Absolute High Ref value.

---

**NOTE.** *this command affects the associated reference level parameter for all MEASurements:IMMed and the eight periodic measurements. To change the parameter for individual measurements, use the MEASUrement:MEAS<x>:REFLevel commands.*

---

**Group** Measurement

**Syntax** MEASUrement:REFLevel:ABSolute:HIGH <NR3>  
 MEASUrement:REFLevel:ABSolute:HIGH?

**Related Commands** [MEASUrement:REFLevel:METHOd](#), [MEASUrement:IMMed:TYPE](#),  
[MEASUrement:MEAS<x>:TYPE](#)

**Arguments** <NR3> is the high reference level, in volts. The default is 0.0 V.

**Examples**     `MEASUREMENT:REFLEVEL:ABSOLUTE:HIGH 1.71` sets the high reference level to 1.71 V.

`MEASUREMENT:REFLEVEL:ABSOLUTE:HIGH?` might return  
`:MEASUREMENT:REFLEVEL:ABSOLUTE:HIGH 1.7100E+00` indicating that the absolute high reference level is set to 1.71 V.

## MEASUrement:REFLevel:ABSolute:LOW

This command sets or returns the low reference level, and is the lower reference level when [MEASUrement:REFLevel:METHod](#) is set to Absolute. This command affects the results of rise and fall measurements and is equivalent to selecting Reference Levels from the Measure menu, and then entering the Absolute Low Ref value.

---

**NOTE.** *this command affects the associated reference level parameter for all MEASurements:IMMed and the eight periodic measurements. To change the parameter for individual measurements, use the MEASUrement:MEAS<x>:REFLevel commands.*

---

**Group**     Measurement

**Syntax**     `MEASUrement:REFLevel:ABSolute:LOW <NR3>`  
`MEASUrement:REFLevel:ABSolute:LOW?`

**Related Commands**     [MEASUrement:REFLevel:METHod](#), [MEASUrement:IMMed:TYPE](#),  
[MEASUrement:MEAS<x>:TYPE](#)

**Arguments**     <NR3> is the low reference level, in volts. The default is 0.0 V.

**Examples**     `MEASUREMENT:REFLEVEL:ABSOLUTE:LOW 0.0` sets the low reference level to 0.0 V.

`MEASUREMENT:REFLEVEL:ABSOLUTE:LOW?` might return  
`:MEASUREMENT:REFLEVEL:ABSOLUTE:LOW 0.0000E+00` indicating that the absolute low reference level is set to 0.0 V.

## MEASUrement:REFLevel:ABSolute:MID[1]

This command sets or returns the mid reference level, and is the 50% reference level when [MEASUrement:REFLevel:METHod](#) is set to Absolute. This command affects the results of period, frequency, delay, and all cyclic measurements and

is equivalent to selecting Reference Levels from the Measure menu, and then entering the Absolute Mid Ref value.

---

**NOTE.** *this command affects the associated reference level parameter for all MEASurements:IMMed and the eight periodic measurements. To change the parameter for individual measurements, use the MEASurement:MEAS<x>:REFLevel commands.*

---

**Group** Measurement

**Syntax** MEASurement:REFLevel:ABSolute:MID[1] <NR3>  
MEASurement:REFLevel:ABSolute:MID[1]?

**Related Commands** [MEASurement:REFLevel:METHod](#)

**Arguments** <NR3> is the mid reference level, in volts. The default is 0.0 V.

**Examples** MEASUREMENT:REFLEVEL:ABSOLUTE:MID1 .71 sets the mid reference level to .71 V.

MEASUREMENT:REFLEVEL:ABSOLUTE:MID? might return  
:MEASUREMENT:REFLEVEL:ABSOLUTE:MID 0.7100E+00 indicating  
that the absolute mid1 reference level is set to .71 V.

## MEASurement:REFLevel:ABSolute:MID2

This command sets or returns the mid reference level for the "to" waveform when taking a delay measurement, and is the 50% reference level when [MEASurement:REFLevel:METHod](#) is set to Absolute. This command affects the results of delay measurements and is equivalent to selecting Reference Levels from the Measure menu, and then entering the Absolute Mid2 Ref value.

---

**NOTE.** *this command affects the associated reference level parameter for all MEASurements:IMMed and the eight periodic measurements. To change the parameter for individual measurements, use the MEASurement:MEAS<x>:REFLevel commands.*

---

**Group** Measurement

**Syntax**      `MEASUREMENT:REFLEVEL:ABSOLUTE:MID2 <NR3>`  
`MEASUREMENT:REFLEVEL:ABSOLUTE:MID2?`

**Related Commands**      [MEASUREMENT:REFLEVEL:METHOD](#)

**Arguments**      <NR3> is the mid reference level, in volts. The default is 0.0 V.

**Examples**      `MEASUREMENT:REFLEVEL:ABSOLUTE:MID2 0.5` sets the mid reference level for the delay waveform to 0.5 V.

`MEASUREMENT:REFLEVEL:ABSOLUTE:MID2?` might return  
`:MEASUREMENT:REFLEVEL:ABSOLUTE:MID2 0.5000E+00` indicating that the absolute mid2 reference level is set to 0.5 V.

## MEASUREMENT:REFLEVEL:METHOD

This command specifies or queries the reference level units used for measurement calculations. This command is equivalent to selecting Reference Levels from the Measure menu, and then choosing the desired reference level from the Units group box.

---

**NOTE.** *this command affects the associated reference level parameter for all MEASUREMENTS:IMMED and the eight periodic measurements. To change the parameter for individual measurements, use the MEASUREMENT:MEAS<x>:REFLEVEL commands.*

---

**Group**      Measurement

**Syntax**      `MEASUREMENT:REFLEVEL:METHOD {ABSOLUTE|PERCENT}`  
`MEASUREMENT:REFLEVEL:METHOD?`

**Arguments**      **ABSOLUTE** specifies that the reference levels are set explicitly using the `MEASUREMENT:REFLEVEL:ABSOLUTE` commands. This method is useful when precise values are required (for example, when designing to published interface specifications, such as RS-232-C).

**PERCENT** specifies that the reference levels are calculated as a percent relative to HIGH and LOW. The percentages are defined using the `MEASUREMENT:REFLEVEL:PERCENT` commands.

**Examples**      `MEASUREMENT:REFLEVEL:METHOD ABSOLUTE` specifies that explicit user-defined values are used for the reference levels.

MEASUREMENT:REFLEVEL:METHOd? might return  
:MEASUREMENT:REFLEVEL:METHOd PERCENT indicating that  
the reference level units used are calculated as a percent relative to HIGH and  
LOW.

## MEASUrement:REFLevel:PERCent:HIGH

This command sets or returns the percent (where 100% is equal to HIGH) that is used to calculate the high reference level when [MEASUrement:REFLevel:METHod](#) is set to Percent. This command affects the results of rise and fall measurements and is equivalent to selecting Reference Levels from the Measure menu, and then entering the Percentage High Ref value.

---

**NOTE.** *this command affects the associated reference level parameter for all MEASurements:IMMed and the eight periodic measurements. To change the parameter for individual measurements, use the MEASUrement:MEAS<x>:REFLevel commands.*

---

<b>Group</b>	Measurement
<b>Syntax</b>	MEASUrement:REFLevel:PERCent:HIGH <NR3> MEASUrement:REFLevel:PERCent:HIGH?
<b>Related Commands</b>	<a href="#">MEASUrement:REFLevel:METHod</a> , <a href="#">MEASUrement:IMMed:TYPe</a> , <a href="#">MEASUrement:MEAS&lt;x&gt;:TYPe</a>
<b>Arguments</b>	<NR3> is the high reference level, ranging from 0 to 100%. The default high reference level is 90%.
<b>Examples</b>	MEASUREMENT:REFLEVEL:PERCENT:HIGH 95 sets the high reference level to 95% of HIGH.  MEASUREMENT:REFLEVEL:PERCENT:HIGH? might return :MEASUREMENT:REFLEVEL:PERCENT:HIGH 90 indicating that the percentage high reference level is set to 90% of HIGH.

## MEASUrement:REFLevel:PERCent:LOW

This command sets or returns the percent (where 100% is equal to HIGH) that is used to calculate the low reference level when [MEASUrement:REFLevel:METHod](#) is set to Percent. This command affects the

results of rise and fall measurements and is equivalent to selecting Reference Levels from the Measure menu, and then entering the Percentage Low Ref value.

---

**NOTE.** *this command affects the associated reference level parameter for all MEASurements:IMMed and the eight periodic measurements. To change the parameter for individual measurements, use the MEASurement:MEAS<x>:REFLevel commands.*

---

<b>Group</b>	Measurement
<b>Syntax</b>	MEASurement:REFLevel:PERCent:LOW <NR3> MEASurement:REFLevel:PERCent:LOW?
<b>Related Commands</b>	MEASurement:REFLevel:METHod, MEASurement:IMMed:TYPE, MEASurement:MEAS<x>:TYPE
<b>Arguments</b>	<NR3> is the low reference level, ranging from 0 to 100%. The default low reference level is 10%.
<b>Examples</b>	MEASUREMENT:REFLEVEL:PERCENT:LOW 15 sets the high reference level to 15% of HIGH.  MEASUREMENT:REFLEVEL:PERCENT:LOW? might return :MEASUREMENT:REFLEVEL:PERCENT:LOW 10 indicating that the percentage high reference level is set to 10% of HIGH.

## MEASurement:REFLevel:PERCent:MID[1]

This command sets or returns the percent (where 100% is equal to HIGH) that is used to calculate the mid reference level when MEASurement:REFLevel:METHod is set to Percent. This command affects the results of period, frequency, delay, and all cyclic measurements and is equivalent to selecting Reference Levels from the Measure menu and, then entering the Percentage Mid Ref value.

---

**NOTE.** *this command affects the associated reference level parameter for all MEASurements:IMMed and the eight periodic measurements. To change the parameter for individual measurements, use the MEASurement:MEAS<x>:REFLevel commands.*

---

<b>Group</b>	Measurement
--------------	-------------

**Syntax**     `MEASUREMENT:REFLevel:PERCENT:MID[1] <NR3>`  
`MEASUREMENT:REFLevel:PERCENT:MID[1]?`

**Related Commands**     [MEASUREMENT:REFLevel:METHod](#)

**Arguments**     <NR3> is the mid reference level, ranging from 0 to 100%. The default mid reference level is 50%.

**Examples**     `MEASUREMENT:REFLEVEL:PERCENT:MID1 60` sets the mid reference level to 60% of HIGH.

`MEASUREMENT:REFLEVEL:PERCENT:MID?` might return  
`:MEASUREMENT:REFLEVEL:PERCENT:MID 65` indicating that the percentage mid reference level is set to 65% of HIGH.

## MEASUREMENT:REFLevel:PERCENT:MID2

This command sets or returns the percent (where 100% is equal to HIGH) that is used to calculate the mid reference level for the second waveform specified when [MEASUREMENT:REFLevel:METHod](#) is set to Percent. This command affects the results of delay measurements and is equivalent to selecting Reference Levels from the Measure menu, and then entering the Percentage Mid2 Ref value.

---

**NOTE.** *this command affects the associated reference level parameter for all MEASUREMENTS:IMMed and the eight periodic measurements. To change the parameter for individual measurements, use the MEASUREMENT:MEAS<x>:REFLevel commands.*

---

**Group**     Measurement

**Syntax**     `MEASUREMENT:REFLevel:PERCENT:MID2 <NR3>`  
`MEASUREMENT:REFLevel:PERCENT:MID2?`

**Related Commands**     [MEASUREMENT:REFLevel:METHod](#)

**Arguments**     <NR3> is the mid reference level, ranging from 0 to 100%. The default mid reference level is 50%.

**Examples**     `MEASUREMENT:REFLEVEL:PERCENT:MID2 40` sets the mid2 reference level to 40% of HIGH.

MEASUREMENT:REFLEVEL:PERCENT:MID2? might return  
:MEASUREMENT:REFLEVEL:PERCENT:MID2 45 indicating that the  
percentage mid2 reference level is set to 45% of HIGH.

## MEASUrement:SNAPShot (No Query Form)

Displays the measurement snapshot list on the oscilloscope screen, which is a list of the immediate values of all available measurements for the active signal.

**Group** Measurement

**Syntax** MEASUrement:SNAPShot

### Examples

## MEASUrement:STATIstics:MODE

This command controls the operation and display of management statistics. This command is equivalent to selecting Measurement Setup from the Measure menu, clicking the Statistics button and then choosing the desired Measurement Format.

**Group** Measurement

**Syntax** MEASUrement:STATIstics:MODE {OFF|ON}  
MEASUrement:STATIstics:MODE?

**Related Commands** [MEASUrement:STATIstics:WEIghting](#)

**Arguments** OFF turns all measurements off. This is the default value.  
ON turns on statistics and displays all statistics for each measurement.

**Examples** MEASUREMENT:STATISTICS:MODE OFF turns statistic measurements off.  
MEASUREMENT:STATISTICS:MODE? might return  
:MEASUREMENT:STATISTICS:MODE ON indicating that  
measurement statistics are turned on and all statistics are being displayed for  
each measurement.



## MEASUrement:STATIstics:WEIghting

This command sets or returns the time constant for mean and standard deviation statistical accumulations. This command is equivalent to selecting Measurement Setup from the Measure menu, clicking the Statistics button and entering the desired Weight n= value.

**Group** Measurement

**Syntax** MEASUrement:STATIstics:WEIghting <NR1>  
MEASUrement:STATIstics:WEIghting?

**Related Commands** [MEASUrement:STATIstics:MODE](#)

**Arguments** <NR1> is the time constant for the mean and standard deviation statistical accumulations.

**Examples** MEASUREMENT:STATISTICS:WEIGHTING 4 sets statistical weighting to 4.  
  
MEASUREMENT:STATISTICS:WEIGHTING? might return  
:MEASUREMENT:STATISTICS:WEIGHTING 4 indicating that  
measurement statistics weighting is currently set to 4.

## MESSAge:BOX

Defines the size and position of the message window. This command does not display the window unless MESSAge:STATE is ON.

**Group** Display

**Syntax** MESSAge:BOX <X1>,<X2>,<Y1>,<Y2>  
MESSAge:BOX?

**Arguments** <X1> and <X2> = 0 to 639, and are pixel positions along the horizontal axis. <X1>defines the left and <X2>defines the right side of the window.  
  
<Y1> and <Y2> = 0 to 479, and are pixel positions along the vertical axis. <Y1> defines the top and <Y2> defines the bottom of the window. The reserved height of all characters is 16 pixels so the window must be at least that high to fully display characters.

## MESSage:CLEAR (No Query Form)

Removes the message text from the message window.

**Group**     Display

**Syntax**    MESSAGE: CLEAR

**Examples**    MESSAGE: CLEAR removes the message from the message box.

## MESSage:SHOW (No Query Form)

Clears the contents of the message window and displays the new message in the window.

**Group**     Display

**Syntax**    MESSAGE: SHOW <QString>

**Arguments**    <QString> is the message and can include any of the characters shown in the TDS Character Chart in Appendix A as well as characters from the Cyrillic, Korean, Japanese, and Chinese character sets. The maximum length of the message is 1000 characters; longer strings are ignored.

The message area is the height and width you have set using the MESSAGE:BOX command. The length of the message that fits in the message area depends on the contents of the message because the width of characters varies.

If the message exceeds the limits of the message box, either horizontally or vertically, the portion of the message that exceeds the limits will not be displayed. The message string itself is not altered. The entire message can be returned as a query response regardless of what is displayed in the message box.

The message is left-justified, and is displayed on a single line starting with the top most line in the window. A new line character can be embedded in the string to position the message on multiple lines. You can also use white space and tab characters to position the message within a line. Text which does not fit within the message box is truncated. Defining a message box text string erases any previously displayed text within the message box.

You can send a tab by transmitting a tab character (\t or \x09) followed by two characters representing the most significant eight bits followed by the least significant eight bits of a 16-bit number. The number specifies the pixel column

relative to the left margin of the message area. For example, to tab to pixel 13, send TAB (\t or \x09), NUL (decimal 0), and CR (decimal 13).

For example, using hexadecimal escape sequences, MESSAGE:SHOW '\x09\x01\x17Hello' when sent as a command would cause the 'Hello' to be displayed starting at pixel position 279 relative to the left margin set by the MESSAGE:BOX command. If you want to display characters starting at position 279, then  $279 = 0x0117$ ; split the hexadecimal number into two characters 0x01 and 0x17 and send \x09\x01\x17.

Special characters which control decoration are two character sequences where the first character is an escape (0x1b) and the second character is as follows:

- Bit 7 — 0
- Bit 6 — If set, inverse video is toggled from current state and the following text is displayed in the new inverse state until the state is toggled again. Remaining bits are ignored.
- Bit 5 — If set, the color index in the four LSB's (bits 0 through 3) is applied to the foreground or background color depending on the fg/bg bit (bit 4).
- Bit 4 — If set, color change is applied to the background, otherwise applies to the foreground.
- Bit 0-3 — Specifies the color index (0 through 15) to change color to as follows:
  - Index 0 Black (background)
  - Index 1 Yellow (Ch1)
  - Index 2 Cyan (Ch2)
  - Index 3 Magenta (Ch3)
  - Index 4 Green (Ch4)
  - Index 5 Red (math)
  - Index 6 White (reference)
  - Index 7 Orange
  - Index 8 Gray (graticule)
  - Index 9 White (text)
  - Index 10 Tek Blue
  - Index 11 Bright Blue
  - Index 12 undefined
  - Index 13 Blue
  - Index 14 undefined
  - Index 15 Dark Blue

The ESC (escape) character followed by the @ character turns inverse video on or off and can be embedded in the message string. Example: "abcESC@defESC@ghi" specifies the string "abcdefghi" where the "def" portion is displayed in inverse video.

Example: "abcESC#defESC)ESC@ghi" specifies the string "abcdefghi" where the "def" portion appears in the channel 3 color (magenta) and the "ghi" portion appears in the normal text color except it's in inverse video.

An alternate way to enter characters is octal escape sequences. This consists of a backslash followed by numerals in the standard C language printf fashion.

Another way to enter characters is `\xnn` where the `nn` is the hexadecimal value of the character to display.

An advantage of these methods is that any controller program can be used. Another advantage is it's easy to access characters with the high bit set, that is, those characters with a decimal value greater than 127.

An alternate way to enter certain characters is with a backslash followed by a single character (following "standard" Unix) as follows:

- `n` newline (carriage return and linefeed)
- `\` backslash (`\` is required to get a backslash character)
- `j` Japanese character encoding, the next two characters make one hexadecimal JIS value as explained later
- `t` horizontal tab; the next 2 characters specify the pixel column to tab to as explained earlier
- `u` Unicode encoding, the next two characters make one hexadecimal Unicode value as explained later

If a backslash is followed by an undefined special character, the backslash is ignored and the character following it is accepted as is.

---

**NOTE.** *The use of any escape codes other than those described above may produce unpredictable results.*

---

To specify Russian, Chinese, and Korean, use the 16 bit Unicode value characters as shown in the example disk files. For additional information see *The Unicode Standard, Version 2.0*. Only characters used in the oscilloscope are available. Unicode values are represented in a quoted string as `"\uxxxyy"` where `xx` and `yy` are both hexadecimal values and `xx` and `yy` are concatenated to form a single hexadecimal number. For example, `"\u0413"` is the Cyrillic capital letter GHE and 0413 is the Unicode value for that character. English and European characters can also be specified using the `\u` Unicode sequence. In that case `xx` is simply 00 and `yy` is the hexadecimal equivalent of the 8 bit character value.

To specify Japanese characters, use the 16 bit Japanese Interchange Standard (JIS X 0208-1983) value. JIS values are represented in a quoted string as `"\jxxxyy"` where `xx` and `yy` are both hexadecimal values and `xx` and `yy` are concatenated to form a single hexadecimal number.

Supported Unicode and JIS characters are shown in the CHARSETS directory of the disk which accompanies this manual.

**Examples** MESSAGE:SHOW "Hello world" displays "Hello world" in the upper left corner of the box (you can define the box size with the MESSAGE BOX command).

MESSAGE:SHOW "Hello ◀@world◀@ ... hello" displays "Hello world ... hello" in the upper left corner of the box and the word "world" is displayed in inverse video. In this example, a ◀ stands for the escape character. The escape character may appear differently for you depending on your controller program.

MESSAGE:SHOW '\x09\x00\xa0\033@Hello world\033@n\nYou can display messages on your \033\*\204\033) oscilloscope\nusing any character or \033\$color\033) used by the\noscilloscope. \033!A \033"few \033%examples \033#follow\033):\n\n\021\027\214\232\340\341\342\343\344 \u042a\u042e\u042f\u3080\u3084\u8017\n\n\033\*\205\n\033%\206' displays the message as shown below. "Color" and "A few examples follow" are displayed in green, yellow, cyan, red, and magenta respectively; Tek and Tektronix are also in Tektronix' colors.

## MESSage:STATE

Controls the display of the message window.

**Group** Display

**Syntax** MESSAGE:STATE {ON|OFF|<NR1>}  
MESSAGE:STATE?

**Related Commands** [MESSage:BOX](#), [MESSage:SHOW](#)

**Arguments** OFF or <NR1>= 0 removes the message window from the display.  
ON or <NR1> ≠ 0 displays the message window and its contents on the display.

## NEWpass (No Query Form)

This command (no query form) changes the password that enables access to password protected data. The PASSWord command must be successfully executed before using this command or an execution error will be generated.

**Group** Miscellaneous

**Syntax** NEWpass <QString>

**Related Commands**    [\\*PUD](#)

**Arguments**    <QString> is the new password, which can contain up to 16 characters.

**Examples**    NEWPASS "mypassword" creates a new password (mypassword) for accessing your protected data.

## \*OPC

This command generates the operation complete message in the Standard Event Status Register (SESR) when all pending commands that generate an OPC message are complete. The \*OPC? query places the ASCII character "1" into the output queue when all such OPC commands are complete. The \*OPC? response is not available to read until all pending operations finish. For a complete discussion of the use of these registers and the output queue, see Registers and Queues.

The \*OPC command allows you to synchronize the operation of the instrument with your application program. For more information, see Synchronization Methods.

**Table 2-36: Commands the Generate an OPC Message**

Operation	Command
Single sequence acquisition	<a href="#">ACQuire:STATE ON</a> or <a href="#">ACQuire:STATE RUN</a> or <a href="#">ACQuire:STATE 1</a> (when <a href="#">ACQuire:STOPAfter</a> is set to SEquence)
Hard copy operation	<a href="#">HARDCopy START</a>
Calibration step	Refer to the optional Service Manual.

**Group**    Status and Error

**Syntax**    [\\*OPC](#)  
              [\\*OPC?](#)

**Related Commands**    [BUSY?](#), [\\*WAI](#)

**Examples**    \*OPC generates the operation complete message in the SESR at the completion of all pending OPC operations.

\*OPC? might return 1 to indicate that all pending OPC operations are finished.

## \*OPT? (Query Only)

This query-only command returns a list of the options installed in your instrument.

<b>Group</b>	Status and Error
<b>Syntax</b>	*OPT?
<b>Examples</b>	*OPT? might return :3M:16 Meg Max 4M/CH

## \*PSC

This command sets and queries the power-on status flag that controls the automatic power-on handling of the DESER, SRER, and ESER registers. When \*PSC is true, the DESER register is set to 255 and the SRER and ESER registers are set to 0 at power-on. When \*PSC is false, the current values in the DESER, SRER, and ESER registers are preserved in nonvolatile memory when power is shut off and are restored at power-on.

<b>Group</b>	Status and Error
<b>Syntax</b>	*PSC <NR1> *PSC?
<b>Related Commands</b>	<a href="#">DESE</a> , <a href="#">*ESE</a> , <a href="#">FACtory</a> , <a href="#">*RST</a> , <a href="#">*SRE</a>
<b>Arguments</b>	<NR1> = 0 sets the power-on status clear flag to false, disables the power-on clear and allows the instrument to possibly assert SRQ after power-on; any other value sets the power-on status clear flag to true, enabling the power-on status clear and prevents any SRQ assertion after power on.
<b>Examples</b>	*PSC 0 sets the power-on status clear flag to false. *PSC? might return 1 to indicate that the power-on status clear flag is set to true.

## \*PUD

This command sets or returns a string of Protected User Data. This data is protected by the PASSWord command. You can modify it only by first entering the correct password. This password is not necessary to query the data.



<b>Group</b>	Status and Error
<b>Syntax</b>	<code>*PUD {&lt;Block&gt; &lt;QString&gt;}</code> <code>*PUD?</code>
<b>Arguments</b>	<code>&lt;Block&gt;</code> is a block containing up to 300 characters. <code>&lt;QString&gt;</code> is a string containing up to 300 characters.
<b>Examples</b>	<code>*PUD #229This instrument belongs to me</code> stores the string "This instrument belongs to me" in the user protected data area. <code>*PUD?</code> might return <code>#221PROPERTY OF COMPANY X</code>

## **\*RCL (No Query Form)**

This command (no query form) restores the state of the instrument from a copy of the settings stored in memory (The settings are stored using the `*SAV` command). If 'factory' is referenced (by specifying '0'), the factory default values will be restored.

This command is equivalent to `RECALL:SETup` and performs the same function as selecting Recall from the File menu, and then pressing the Setup button.

<b>Group</b>	Save and Recall
<b>Syntax</b>	<code>*RCL &lt;NR1&gt;</code>
<b>Related Commands</b>	<a href="#">FACtory</a> , <a href="#">*LRN?</a> , <a href="#">RECALL:SETUp</a> , <a href="#">*RST</a> , <a href="#">*SAV</a> , <a href="#">SAVE:SETUp</a>
<b>Arguments</b>	<code>&lt;NR1&gt;</code> is a value in the range from 0 to 10. A value of 1 to 10, specifies a saved setup storage location. Specifying a value of 0 causes the instrument factory defaults to be restored. Using an out of range value causes an error (222, "Data out of range").
<b>Examples</b>	<code>*RCL 3</code> restores the instrument from a copy of the settings stored in memory location 3.

## RECALL:SETUp (No Query Form)

This command (no query form) returns stored or factory settings to the instrument from a copy of the settings stored in memory. If factory is referenced (by specifying '0'), the factory default values will be restored. This command is equivalent to \*RCL and performs the same function as selecting Recall from the File menu, and then choosing the Setup button.

**Group** Save and Recall

**Syntax** RECALL:SETUp {FACTory|<NR1>|<file path>}

**Related Commands** [FACTory](#), [\\*RCL](#), [\\*RST](#), [\\*SAV](#), [SAVE:SETUp](#)

**Arguments** FACTory restores the factory setup.

<NR1> is a value in the range from 0 to 10. A value of 1 to 10, specifies a saved setup storage location. Specifying a value of 0 causes the instrument factory defaults to be restored. Using an out of range value causes an error (222, "Data out of range").

<file path> specifies a location for an instrument setup file. <file path> is a quoted string that defines the file name and path. Input the file path using the form <drive>:<dir>/<filename>.<drive> and one or more <dir>s are optional. If you do not specify them, the instrument will read the file from the default directory (C:\TekScope\setups). <filename> stands for a filename of up to 128 characters (use of wildcard characters in filenames is not supported). Filename extensions are not required, but highly recommended.

**Examples** RECALL:SETUP FACTORY recalls (and makes current) the instrument setup to its factory defaults.

RECALL:SETUP 2 recalls the instrument setup from setup storage location 2.

RECALL:SETUP "TEK00000.SET" recalls the setup from the file TEK00000.SET in the default directory for setups (C:\TekScope\setups).

## RECALL:WAVEform (No Query Form)

This command (no query form) recalls a stored waveform to a reference location. This command is equivalent to selecting Recall from the File menu, and then pressing the Waveform button.

**Group** Save and Recall

**Syntax**     `RECALL:WAVEform <file path>,REF<x>`

**Related Commands**     [SAVE:WAVEform](#)

**Arguments**     REF<x> specifies a location in internal reference memory. Reference memory location values range from 1 through 4.

                     <file path> specifies a location for a stored waveform file. <file path> is a quoted string that defines the file name and path. Input the file path using the form <drive>:<dir>/<filename>.<drive> and one or more <dir>s are optional. If you do not specify them, the instrument will read the waveform from the default directory (C:\TekScope\Waveforms). <filename> stands for a filename of up to 128 characters (use of wildcard characters in filenames is not supported). Filename extensions are not required but are highly recommended.

**Examples**     `RECALL:WAVEFORM "TEK00000.WFM",REF1` recalls the waveform stored in the file named TEK00000.WFM from the default directory for waveforms (C:\TekScope\Waveforms) to the reference location 1.

## REF<x>:DATE? (Query Only)

This query-only command returns the date that a reference waveform was stored.

**Group**     Vertical

**Syntax**     `REF<x>:DATE?`

## REF<x>:HORizontal:POSition

This command sets or returns the horizontal display position of the reference waveform. The reference waveform is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is equivalent to selecting Reference Waveforms from the File menu, choosing Reference Setup¼ from the drop-down list, selecting a reference waveform, and then entering the horizontal position value using the multipurpose knob.

**Group**     Vertical

**Syntax**    `REF<x>:HORizontal:POSition <NR3>`  
               `REF<x>:HORizontal:POSition?`

**Arguments**    `<NR3>` argument specifies the horizontal position of the specified reference waveform in percentage of the waveform that is displayed to the right of the center vertical graticule. The range of this argument is from 0 through 100.

**Examples**    `REF2:HORIZONTAL:POSITION 0.0` positions the start of the Reference 2 waveform at the center graticule.

`REF1:HORIZONTAL:POSITION?` might return `:REF1:HORIZONTAL:POSITION 50` indicating that the Reference 1 waveform is centered on the horizontal graticule.

## REF<x>:HORizontal:SCALE

This command sets or returns the horizontal scale for a reference waveform.

**Group**    Vertical

**Syntax**    `REF<x>:HORizontal:SCALE <NR3>`  
               `REF<x>:HORizontal:SCALE?`

## REF<x>:TIME? (Query Only)

This query-only command returns the time that a reference waveform was stored.

**Group**    Vertical

**Syntax**    `REF<x>:TIME?`

## REF<x>:VERTical:POSITION

This command sets or returns the vertical position of a reference waveform. The reference waveform is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments.

This command is equivalent to selecting Reference Waveforms from the File menu, choosing Reference Setup from the drop-down list, selecting a reference waveform, and then entering the Position value using the multipurpose knob.

Increasing the position value of a waveform causes the waveform to move up, and decreasing the position value causes the waveform to move down. Position adjusts only the display position of a waveform. The position value determines the vertical graticule coordinate at which signal values are displayed. For example, if the position for Reference 3 is set to 2.0, the signal represented by that reference will be displayed at 2.0 divisions above the center of the screen.

---

**NOTE.** *References are static. All position and scale changes are applied in a post-processing mode.*

---

<b>Group</b>	Vertical
<b>Syntax</b>	REF<x>:VERTical:POSition <NR3> REF<x>:VERTical:POSition?
<b>Related Commands</b>	<a href="#">CH&lt;x&gt;:POSition</a> , <a href="#">MATH[1]:VERTical:POSition</a>
<b>Arguments</b>	<NR3> is the desired position, in divisions from the center horizontal graticule. The range is from -5 to 5 divisions.
<b>Examples</b>	REF2:VERTICAL:POSITION 1.3E+00 positions the Reference 2 input signal 1.3 divisions above the center horizontal graticule.  REF1:VERTICAL:POSITION? might return :REF1:VERTICAL:POSITION -1.3000E+00 indicating that the current position of Reference 1 is 1.3 divisions below the center horizontal graticule.

## REF<x>:VERTical:SCAle

This command sets or returns the vertical scale of a reference waveform. The reference waveform is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is equivalent to selecting Reference Waveforms from the File menu, choosing Reference Setup from the drop-down list, selecting a reference waveform, and then entering the Scale value using the multipurpose knob.

Each waveform has a vertical scale parameter. For a signal with constant amplitude, increasing the Scale causes the waveform to be displayed smaller. Decreasing the scale causes the waveform to be displayed larger.

Scale affects all waveforms, but affects reference and math waveforms differently from channel waveforms:

- For reference and math waveforms, this setting controls the display only, graphically scaling these waveforms and having no affect on the acquisition hardware.
- For channel waveforms, this setting controls the vertical size of the acquisition window as well as the display scale. The range and resolution of scale values depends on the probe attached and any other external factors you have specified.

**Group** Vertical

**Syntax** REF<x>:VERTical:SCAle <NR3>  
REF<x>:VERTical:SCAle?

**Related Commands** [CH<x>:SCAle](#), [MATH\[1\]:VERTical:SCAle](#)

**Arguments** <NR3> is the gain in user units per division.

**Examples** REF4:VERTICAL:SCALE 100E-03 sets the Reference 4 scale to 100 mV per division.

REF4:VERTICAL:SCALE? might return :REF2:VERTICAL:SCALE  
1.0000e+00 indicating that the current vertical scale setting for Reference 2 is 1 V per division.

## REM (No Query Form)

This command (no query form) embeds a comment within GPIB programs as a means of internally documenting the programs. The instrument ignores these embedded comment lines.

**Group** Miscellaneous

**Syntax** REM <QString>

**Related Commands** [NEWpass](#), [\\*PUD](#)

**Arguments** <QString> is a string that can contain a maximum of 80 characters.

**Examples**     REM "This is a comment" is a comment string that the instrument will ignore.

## \*RST (No Query Form)

This command (no query form) resets the instrument to the factory default settings. The \*RST command does not alter the following:

- The state of the IEEE Std 488.1-1987 interface
- The selected IEEE Std 488.1-1987 address of the instrument
- Calibration data that affect device specifications
- The Output Queue
- The Service Request Enable Register setting
- The Power-on status clear flag setting
- Alias definitions
- Stored settings
- The \*PUD? Response
- Any of the values associated with the DATA command.

This command is equivalent to pressing the **DEFAULT SETUP** button on the front panel.

**Group**     Status and Error

**Syntax**     \*RST

**Related Commands**     [FACtory](#), [RECALL:SETUp](#), [SAVe:SETUp](#)

**Arguments**     None

**Examples**     \*RST resets the instrument settings to factory defaults.

## \*SAV (No Query Form)

This command (no query form) stores the state of the instrument to a specified memory location. You can later use the \*RCL command to restore the instrument to this saved state. \

This command is equivalent to selecting Save As from the File menu, and then pressing the Setup button.

**Group** Save and Recall

**Syntax** \*SAV <NR1>

**Related Commands** [\\*RCL](#), [RECALL:SETUp](#), [SAVe:SETUp](#)

**Arguments** <NR1> specifies a location in which to save the state of the instrument. Location values range from 1 through 10. Using an out-of-range location value causes an execution error. Any settings that have been stored previously at this location will be overwritten.

**Examples** \*SAV 2 saves the current instrument state in memory location 2.

## SAVe:IMAGe (No Query Form)

Saves a capture of the screen image into the specified file. Supported image formats are PNG, Windows Bitmap and TIFF. If an extension for a supported file type is added to the file name, then the corresponding format will be used. If no supported extension is added to the file, the format to use will be determined by the value obtained from the :SAVe:IMAGe:FILEFormat? query.

**Group** Save and Recall

**Syntax** SAvE:IMAGe <file path>

**Arguments** <file path> is a filename, including path, where the image will be saved.

## SAVe:IMAGe:FILEFormat

Sets the file format to use for saving screen images when the file type cannot be determined from the given file name or when screen images are captured by using the front panel.

**Group** Save and Recall



**Syntax**     `SAVE:IMAGE:FILEFormat {PNG|BMP|TIFF}`  
`SAVE:IMAGE:FILEFormat?`

**Arguments**     PNG saves the file in the Portable Network Graphics format.  
                     BMP saves the file in the Windows bitmap format.  
                     TIFF saves the file in the Tag Image File Format.

## SAVE:SETUp (No Query Form)

This command (no query form) stores the state of the instrument to a specified memory location. You can later use the \*RCL command to restore the instrument to this saved state. This is equivalent to selecting Save As from the File menu, and then pressing the Setup button.

**Group**     Save and Recall

**Syntax**     `SAVE:SETUp {<file path>|<NR1>}`

**Related Commands**     [\\*RCL](#), [RECALL:SETUp](#), [\\*SAV](#)

**Arguments**     <file path> is the target location for storing the setup file. <file path> is a quoted string that defines the file name and path. Input the file path using the form <drive>:<dir>/<filename>. <drive> and one or more <dir>s are optional. If you do not specify them, the instrument will store the file in the default directory (C:\TekScope\setups). <filename> stands for a filename of up to 128 characters. (Use of wildcard characters in filenames is not supported.) Filename extensions are not required but are highly recommended. For setups, use the extension ".SET".

                     <NR1> specifies a location for saving the current front-panel setup. The front-panel setup value ranges from 1 to 10. Using an out-of-range value causes an execution error. Any settings that have been stored previously at this location will be overwritten.

**Examples**     `SAVE:SETUp 5` saves the current instrument setup in memory location 5.

`SAVE:SETUp "TEK00000.SET"` saves the current instrument setup in the file TEK00000.SET in the default directory for setups (C:\TekScope\setups).

## SAVe:WAVEform (No Query Form)

This command (no query form) saves a waveform to one of four reference memory locations or a file. This command is equivalent to selecting Reference Waveforms from the File menu and choosing Save Wfm from the drop-down list.

**Group** Save and Recall

**Syntax** SAvE:WAVEform <wfm>,{<file path>|REF<x>}

**Related Commands** [RECALL:WAVEform](#)

**Arguments** <wfm> is the waveform that will be saved. Valid waveforms include CH<x>, MATH<y>, and REF<x>.

REF<x> specifies the location in internal reference memory to where the waveform is stored.

<file path> is the location to where the waveform is stored. <file path> is a quoted string that defines the file name and path. Input the file path using the form <drive>/<dir>/<filename>. <drive> and one or more <dir>s are optional. If you do not specify them, the instrument will read the waveform from the default directory. <filename> stands for a filename of up to 125 characters, followed by a period (".") and the three-character extension. Internal format waveforms use the .wfm extension for spreadsheet format files or the .dat extension for MathCad format files. The instrument will generate an error if you use any other extension for saving a waveform.

**Examples** SAvE:WAVEFORM MATH2,REF2 saves the Math2 waveform in reference memory location2.

SAvE:WAVEFORM MATH1,"TEK0000.WFM" saves the Math1 waveform to the file TEK0000.WFM in the default directory and on the default drive.

## SAVe:WAVEform:FILEFormat

This command specifies or returns the file format for saved waveforms. Waveform header and timing information is included in the resulting file of non-internal formats. The instrument saves DPO waveforms as a 500 x 200 matrix, with the first row corresponding to the most recently acquired data. The values specified by [DATA:START](#) and [DATA:STOP](#) determine the range of waveform data to output. In the event that DATA:STOP value is greater than the current record length, the current record length determines the last output value.

This command is equivalent to selecting Save As from the File menu, clicking the Waveform button, and choosing the desired waveform file format from the Save as type drop-down list.

**Group** Save and Recall

**Syntax** `SAVE:WAVEform:FILEFormat {INTERNAL|MATHCad|SPREADSheet}`  
`SAVE:WAVEform:FILEFormat?`

**Related Commands** [CURVe](#), [DATA](#), [DATA:START](#), [DATA:STOP](#), [SAVE:WAVEform](#),  
[WFMinpre:NR\\_Pt](#), [WFMOutpre:NR\\_Pt?](#)

**Arguments** `INTERNAL` specifies that waveforms are saved in an internal format, using a .wfm filename extension. These files can be recalled as reference waveforms. When this argument is specified, the settings specified via the `DATA:START` and `DATA:STOP` commands have no meaning as the entire waveform is saved.

`MATHCad` specifies that waveforms are saved in MathCad format, using a .dat filename extension. When saving in this format, waveform values are delimited with new lines. MathCad format enables easy import of waveform data into MathCad or MATLAB. For FastAcq waveforms, data is imported as a matrix. For these formats, waveform header and timing information is saved in a separate header file. MathCad format header files use a \_hdr.dat extension.

`MATLab` specifies that waveforms are saved in Matlab format, using a .dat filename extension. When saving in this format, waveform values are delimited with new lines. MATLAB format enables easy import of waveform data into MathCad or MATLAB. For FastAcq waveforms, data is imported as a matrix. For these formats, waveform header and timing information is saved in a separate header file. MATLAB format header files use a \_hdr.dat extension.

`SPREADSHEETCsv` specifies that waveform data is saved in a format that contains comma delimited values. These waveform data files are named using the .csv filename extension. Saving waveforms in CSV format enables spreadsheet programs to import the data.

`SPREADSHEETTxt` specifies that waveform data is saved in a format that contains tab delimited values. These waveform data files are named using the .txt filename extension. Saving waveforms in this format enables spreadsheet programs to import the data.

**Examples** `SAVE:WAVEFORM:FILEFORMAT INTERNAL` specifies that the internal file format is the format used for saving waveforms.

`SAVE:WAVEFORM:FILEFORMAT?` might return `:SAVE:WAVEFORM:FILEFORMAT INTERNAL` indicating that waveforms are saved using the internal format.

## SEARCH:SEARCH<x>:COPY (No Query Form)

This command copies the search criteria to the trigger, or the trigger criteria to the search.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:COPY {SEARCHtotrigger|TRIGgertosearch|UNDO}

**Arguments** SEARCHtotrigger copies the search criteria to the trigger  
TRIGgertosearch copies the trigger criteria to the search  
UNDO undoes the last copy.

## SEARCH:SEARCH<x>:STATE

This command sets the search state to on or off and the query form returns the search state. The returned value is either 0 or 1 for OFF and ON, respectively.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:STATE {<NR1>|OFF|ON}  
SEARCH:SEARCH<x>:STATE?

**Arguments** OFF or <NR1> = 0 sets the search state to off.  
ON or <NR1> ≠ 0 sets the search state to on.

## SEARCH:SEARCH<x>:TOTAL? (Query Only)

This query-only returns the total amount of matches for this search, which may be more than the number of marks placed if the mark limit has been exceeded.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TOTAL?

**Returns** <NR1> is the total number of matches.

## SEARCH:SEARCH<x>:TRIGger:A:BUS

This command sets or queries the serial search type. There are two serial buses B1 and B2. Each can be independently set to one of three serial search types: I2C, CAN or SPI.

<b>Conditions</b>	This command requires a DPO4AUTO or DPO4EMBD application module.
<b>Group</b>	Search
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:BUS {I2C SPI CAN} SEARCH:SEARCH<x>:TRIGger:A:BUS?
<b>Arguments</b>	I2C specifies the Inter-IC bus. SPI specifies the Serial Peripheral Interface bus. CAN specifies the Controller Area Network bus.

## SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:CONDition

This command sets or queries the search condition for CAN search.

<b>Conditions</b>	This command requires a DPO4AUTO application module.
<b>Group</b>	Search
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:CONDition {SOF FRAMEtype IDentifier DATA IDANDDATA EOF ACKMISS} SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:CONDition?
<b>Arguments</b>	SOF specifies a search based on start of frame. FRAMEtype specifies a search based on frame type. IDentifier specifies a search based on frame identifier. DATA specifies a search based on frame data IDANDDATA specifies a search based on frame identifier and data. EOF specifies end of frame. ACKMISS specifies a search based on the missing the ACK field.

## SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:DATa:DIRectioN

This command sets or queries the CAN search condition to be valid on a READ, WRITE or either. This only applies if the search condition is IDentifier.

<b>Conditions</b>	This command requires a DPO4AUTO application module.
<b>Group</b>	Search
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:DATa:DIRectioN {READ WRITE NOCARE} SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:DATa:DIRectioN?
<b>Arguments</b>	READ specifies a read condition. WRITE specifies a write condition. NOCARE specifies either a read or write condition.

## SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:DATa:QUALifier

This command sets or queries the CAN data qualifier. This only applies if the trigger condition is IDANDDATA or DATA.

<b>Conditions</b>	This command requires a DPO4AUTO application module.
<b>Group</b>	Search
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:DATa:QUALifier {LESSThan MOREThan EQUa1 UNEQUa1} SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:DATa:QUALifier?
<b>Related Commands</b>	<a href="#">SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:CAN:DATa:VALue</a>
<b>Arguments</b>	LESSThan specifies to search for bus data is less than the specified value specified by <a href="#">SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:CAN:DATa:VALue</a> . MOREThan specifies to search for bus data is more than the specified value specified by <a href="#">SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:CAN:DATa:VALue</a> . EQUa1 specifies to search for bus data is equal to the specified value specified by <a href="#">SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:CAN:DATa:VALue</a> .

UNEQUAL specifies to search for bus data is not equal to the specified value specified by [SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:CAN:DATA:VALUE](#).

## SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:CAN:DATA:SIZE

This command sets or queries the length of the data string in bytes to be used for CAN search, if search condition is DATA or IDANDDATA.

**Conditions** This command requires a DPO4AUTO application module.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:CAN:DATA:SIZE <NR1>  
SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:CAN:DATA:SIZE?

**Arguments** <NR1> is the data string length in bytes.

## SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:CAN:DATA:VALUE

This command sets or queries the binary data string to be used for CAN search if search condition is ID or IDANDDATA.

**Conditions** This command requires a DPO4AUTO application module.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:CAN:DATA:VALUE <bin>  
SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:CAN:DATA:VALUE?

**Related Commands** [SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:CAN:DATA:QUALIFIER](#)

**Arguments** <bin> is the data in binary format.

## SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:CAN:FRAMETYPE

This command sets or queries the CAN Frame Type to be used if the search condition is FrameType.

**Conditions** This command requires a DPO4AUTO application module.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:FRAMeType  
{DATA|REMOte|ERRor|OVERLoad}  
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN:FRAMeType?

**Arguments** DATA specifies a data frame.  
REMOte specifies a remote frame.  
ERRor specifies an error frame.  
OVERLoad specifies an overload frame.

## SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN{:IDentifier|:ADDReSS}:MODE

This command sets or queries the CAN addressing mode to standard or extended format.

**Conditions** This command requires a DPO4AUTO application module.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN{:IDentifier|:  
ADDReSS}:MODE {STANDARD|EXTended}  
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN{:IDentifier|:  
ADDReSS}:MODE?

**Arguments** STANDARD specifies an 11-bit identifier field.  
EXTended specifies a 29-bit identifier field.

## SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN{:IDentifier|:ADDReSS}:VALue

This command sets or queries the binary address string to be used for CAN search if search condition is ID or IDANDDATA.

**Conditions** This command requires a DPO4AUTO application module.



**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN{:IDentifier|:  
ADDRess}:VALue <bin>  
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:CAN{:IDentifier|:  
ADDRess}:VALue?

**Arguments** <bin> is the address in binary format.

## SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:ADDRess:MODE

This command sets or queries the I2C address mode to 7 or 10-Bit.

**Conditions** This command requires a DPO4EMBD application module.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:ADDRess:MODE  
{ADDR7|ADDR10}  
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:ADDRess:MODE?

**Arguments** ADDR7 specifies 7-bit addresses.  
ADDR10 specifies 10-bit addresses.

## SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:ADDRess:TYPE

This command sets or queries the I2C address type to I2C special addresses (General Call, HS Mode etc.), or to a user-specified address.

**Conditions** This command requires a DPO4EMBD application module.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:ADDRess:TYPE  
{GENeralcall|STARTbyte|HSmode|EEPROM|USER}  
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:ADDRess:TYPE?

<b>Arguments</b>	<p><b>Generalcall</b> specifies a general call address.</p> <p><b>STARTbyte</b> specifies a start byte address.</p> <p><b>HSmode</b> specifies a high-speed mode address.</p> <p><b>EEPROM</b> specifies an EEPROM address.</p> <p><b>USER</b> specifies a user address.</p>
------------------	--

## SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:ADDResS:VALue

This command sets or queries the binary address string to be used for I2C search if search condition is ADDR or ADDRANDDATA.

<b>Conditions</b>	This command requires a DPO4EMBD application module.
<b>Group</b>	Search
<b>Syntax</b>	<p>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:I2C:ADDResS:VALue &lt;bin&gt;</p> <p>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:I2C:ADDResS:VALue?</p>
<b>Arguments</b>	<bin> is the address in binary format.

## SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:CONDition

This command sets or queries the search condition for I2C search.

<b>Conditions</b>	This command requires a DPO4EMBD application module.
<b>Group</b>	Search
<b>Syntax</b>	<p>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:I2C:CONDition</p> <p>{START STOP REPEATstart ACKMISS ADDRESS DATA ADDRANDDATA}</p> <p>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:BUS:B&lt;x&gt;:I2C:CONDition?</p>
<b>Arguments</b>	<p><b>START</b> specifies a search based on start condition.</p> <p><b>STOP</b> specifies a search based on stop condition.</p> <p><b>REPEATstart</b> specifies a search based on repeat of start condition.</p> <p><b>ACKMISS</b> specifies a search based on missing acknowledgement condition.</p>

ADDRESS specifies a search based on address.

DATA specifies a search based on data.

ADDRANDDATA specifies a search based on address and data.

## SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:DATa:DIRection

This command sets or queries the I2C search condition to be valid on a READ, WRITE or either.

**Conditions** This command requires a DPO4EMBD application module.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:DATa:DIRection  
{READ|WRITE|NOCARE}  
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:DATa:DIRection?

**Arguments** READ specifies a read condition.  
WRITE specifies a write condition.  
NOCARE specifies either a read or write condition.

## SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:DATa:SIze

This command sets or queries the length of the data string in bytes to be used for I2C search, if search condition is DATA or ADDRANDDATA.

**Conditions** This command requires a DPO4EMBD application module.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:DATa:SIze <NR1>  
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:DATa:SIze?

**Arguments** <NR1> is the data string length in bytes.

## SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:DATa:VALue

This command sets or queries the binary data string to be used for I2C search if search condition is DATA or ADDRANDDATA.

**Conditions** This command requires a DPO4EMBD application module.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:DATa:VALue <bin>  
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:DATa:VALue?

**Arguments** <bin> is the data in binary format.

## SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:SPI:CONDition

This command sets or queries the search condition for SPI search.

**Conditions** This command requires a DPO4EMBD application module.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:SPI:CONDition  
{SS|MISO|MOSI|MISOMOSI}  
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:SPI:CONDition?

**Arguments** SS specifies the Slave Selection condition  
MISO specifies the Master-In Slave-Out condition.  
MOSI specifies the Master-Out Slave-In condition.  
MISOMOSI specifies the Master-In Slave-Out and Master-Out Slave-In conditions.

## SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:SPI:DATa{:MISO|:IN}:VALue

This command sets or queries the binary data string to be used for SPI search if search condition is MISO, MISOMOSI.

**Conditions** This command requires a DPO4EMBD application module.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:SPI:DATA{:MISO|:IN}:  
VALue <bin>  
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:SPI:DATA{:MISO|:IN}:  
VALue?

**Arguments** <bin> is the data in binary format.

## SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:SPI:DATA{:MOSI|:OUT}:VALue

This command sets or queries the binary data string to be used for SPI search if search condition is MOSI, or MISOMOSI.

**Conditions** This command requires a DPO4EMBD application module.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:SPI:DATA{:MOSI|:OUT}:  
VALue <bin>  
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:SPI:DATA{:MOSI|:OUT}:  
VALue?

**Arguments** <bin> is the data in binary format.

## SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:SPI:DATA:SIZE

This command sets or queries the length of the data string in bytes to be used for SPI search, if search condition is MISO, MOSI or MISOMOSI.

**Conditions** This command requires a DPO4EMBD application module.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:SPI:DATA:SIZE <NR1>  
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:SPI:DATA:SIZE?

**Arguments** <NR1> is the data string length in bytes.

## SEARCH:SEARCH<x>:TRIGger:A:BUS:SOUrce

This command sets or queries the bus for a serial search.

<b>Conditions</b>	This command requires a DPO4AUTO or DPO4EMBD application module.
<b>Group</b>	Search
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:BUS:SOUrce {B1 B2} SEARCH:SEARCH<x>:TRIGger:A:BUS:SOUrce?
<b>Arguments</b>	B1 specifies the bus 1 source. B2 specifies the bus 2 source.

## SEARCH:SEARCH<x>:TRIGger:A:EDGE:SLOPe

This command sets or returns the slope for an edge search to determine where to place a mark.

<b>Group</b>	Search
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:EDGE:SLOPe {RISe FALL} SEARCH:SEARCH<x>:TRIGger:A:EDGE:SLOPe?
<b>Arguments</b>	RISe specifies a rising edge. FALL specifies a falling edge.

## SEARCH:SEARCH<x>:TRIGger:A:EDGE:SOUrce

This command sets or returns the source waveform for an edge search to determine where to place a mark.

<b>Group</b>	Search
<b>Syntax</b>	SEARCH:SEARCH<x>:TRIGger:A:EDGE:SOUrce {CH1 CH2 CH3 CH4 MATH} SEARCH:SEARCH<x>:TRIGger:A:EDGE:SOUrce?

**Arguments** CH<x> specifies one input channel as the edge source, where <x> = 1, 2, 3 or 4.  
MATH specifies the math waveform as the search source.

## SEARCH:SEARCH<x>:TRIGger:A:LEVel

This command sets or queries the level for an edge search to determine where to place a mark.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:LEVel {<NR3>|ECL|TTL}  
SEARCH:SEARCH<x>:TRIGger:A:LEVel?

**Arguments** <NR3> specifies the trigger level, in volts.  
ECL specifies a -1.3 V trigger level.  
TTL specifies a 1.4 V trigger level.

## SEARCH:SEARCH<x>:TRIGger:A:LEVel:CH<x>

This command sets or queries the level for edge search for the selected channel to determine where to place a mark. Each channel can have an independent level.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:LEVel:CH<x> {<NR3>|ECL|TTL}  
SEARCH:SEARCH<x>:TRIGger:A:LEVel:CH<x>?

**Arguments** <NR3> specifies the trigger level, in volts.  
ECL specifies a -1.3 V trigger level.  
TTL specifies a 1.4 V trigger level.

## SEARCH:SEARCH<x>:TRIGger:A:LEVel:MATH

This command sets or queries the math waveform level for edge search to determine where to place a mark.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:LEVe1:MATH {ECL|TTL}  
SEARCH:SEARCH<x>:TRIGger:A:LEVe1:MATH?

**Arguments** ECL specifies a –1.3 V trigger level.  
TTL specifies a 1.4 V trigger level.

## SEARCH:SEARCH<x>:TRIGger:A:LEVe1:REF<x>

This command sets or queries the reference waveform level for edge search to determine where to place a mark

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:LEVe1:REF<x> {ECL|TTL}  
SEARCH:SEARCH<x>:TRIGger:A:LEVe1:REF<x>?

**Arguments** ECL specifies a –1.3 V trigger level.  
TTL specifies a 1.4 V trigger level.

## SEARCH:SEARCH<x>:TRIGger:A:LOGIc:FUNCTion

This command sets or queries the logic operator for the logic search to determine where to place a mark.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:LOGIc:FUNCTion {AND|NAND|NOR|OR}  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:FUNCTion?

**Arguments** AND specifies to place a mark if all conditions are true.  
NAND specifies to place a mark if any of the conditions are false.  
NOR specifies to place a mark if all conditions are false.  
OR specifies to place a mark if any of the conditions are true.



## SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPut:CH<x>

This command sets or queries the Boolean logic criteria for the logic search to determine where to place a mark.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPut:CH<x> {HIGH|LOW|X}  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPut:CH<x>?

**Arguments** HIGH specifies the logic high.  
LOW specifies the logic low.  
X specifies a "don't care" state.

## SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPut:CLOCK:EDGE

This command sets or queries whether the clock edge is rise or fall for a logic search to determine where to place a mark.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPut:CLOCK:EDGE  
{FALL|RISe}  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPut:CLOCK:EDGE?

**Arguments** RISe specifies a rising edge.  
FALL specifies a falling edge.

## SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPut:CLOCK:SOURce

This command sets or queries the clock source definition for logic search. If a clock source is defined, then the logic search is determined by the state of the other inputs at the clock transition. If no clock source is defined, then the logic search is determined only by the state of the inputs.

**Group** Search

**Syntax**     SEARCH:SEARCH<x>:TRIGger:A:LOGic:INPut:CLOCK:SOURce  
                   {CH1|CH2|CH3|CH4|MATH}  
                   SEARCH:SEARCH<x>:TRIGger:A:LOGic:INPut:CLOCK:SOURce?

**Arguments**     CH<x> specifies a channel input as the clock source, where <x> = 1, 2, 3, or 4.  
                   MATH specifies the math waveform as the clock source.

## SEARCH:SEARCH<x>:TRIGger:A:LOGic:INPut:MATH

This command sets or queries the Boolean logic criteria for the logic search to determine where to place a mark.

**Group**     Search

**Syntax**     SEARCH:SEARCH<x>:TRIGger:A:LOGic:INPut:MATH {HIGH|LOW|X}  
                   SEARCH:SEARCH<x>:TRIGger:A:LOGic:INPut:MATH?

**Arguments**     HIGH specifies a high logic level.  
                   LOW specifies a low logic level.  
                   X specifies a “don’t care” condition.

## SEARCH:SEARCH<x>:TRIGger:A:LOGic:INPut:REF<x>

This command sets or queries the Boolean logic criteria for the logic search to determine where to place a mark.

**Group**     Search

**Syntax**     SEARCH:SEARCH<x>:TRIGger:A:LOGic:INPut:REF<x> {HIGH|LOW|X}  
                   SEARCH:SEARCH<x>:TRIGger:A:LOGic:INPut:REF<x>?

**Arguments**     HIGH specifies a high logic level.  
                   LOW specifies a low logic level.  
                   X specifies a “don’t care” condition.

## SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PATtern:INPut:CH<x>

This command sets or queries the Boolean logic criteria for the logic search to determine where to place a mark.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PATtern:INPut:CH<x>  
{HIGH|LOW|X}  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PATtern:INPut:CH<x>?

**Arguments** HIGH specifies a high logic level.  
LOW specifies a low logic level.  
X specifies a “don’t care” condition.

## SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PATtern:INPut:MATH

This command sets or returns the Boolean logic criteria for the logic search to determine where to place a mark.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PATtern:INPut:MATH  
{HIGH|LOW|X}  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PATtern:INPut:MATH?

**Arguments** HIGH specifies a high logic level.  
LOW specifies a low logic level.  
X specifies a “don’t care” condition.

## SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PATtern:INPut:REF<x>

This command sets or returns the Boolean logic criteria for the logic search to determine where to place a mark.

**Group** Search

**Syntax**     `SEARCH:SEARCH<x>:TRIGger:A:LOGic:PATtern:INPut:REF<x>`  
                   `{HIGH|LOW|X}`  
                   `SEARCH:SEARCH<x>:TRIGger:A:LOGic:PATtern:INPut:REF<x>?`

**Arguments**    `HIGH` specifies a high logic level.  
                   `LOW` specifies a low logic level.  
                   `X` specifies a “don’t care” condition.

## **SEARCH:SEARCH<x>:TRIGger:A:LOGic:PATtern:WHEn**

This command sets or returns the condition for generating a logic pattern search to determine where to place a mark.

**Group**        Search

**Syntax**       `SEARCH:SEARCH<x>:TRIGger:A:LOGic:PATtern:WHEn`  
                   `{TRUE|FALSE|LESSThan|MOREThan|EQUa1|UNEQUa1}`  
                   `SEARCH:SEARCH<x>:TRIGger:A:LOGic:PATtern:WHEn?`

**Arguments**    `TRUE` argument sets the instrument to place a mark when the pattern becomes true.

`FALSE` argument sets the instrument to place a mark when the pattern becomes false.

`LESSThan` argument sets the instrument to place a mark if the specific pattern is true less than the time set by the [SEARCH:SEARCH<x>:TRIGger:A:LOGic:PATtern:WHEn:LESSLimit](#) command.

`MOREThan` argument sets the instrument to place a mark if the specific pattern is true longer than the specified time set by the [SEARCH:SEARCH<x>:TRIGger:A:LOGic:PATtern:WHEn:MORELimit](#) command.

`EQUa1` argument sets the instrument to place a mark if the specific pattern is true longer than the time set by the [SEARCH:SEARCH<x>:TRIGger:A:LOGic:PATtern:WHEn:LESSLimit](#) command, but less than the specified time set by the [SEARCH:SEARCH<x>:TRIGger:A:LOGic:PATtern:WHEn:MORELimit](#) command.

`UNEQUa1` argument sets the instrument to place a mark if the specific pattern is true less than the time set by the [SEARCH:SEARCH<x>:TRIGger:A:LOGic:PATtern:WHEn:LESSLimit](#) command, or longer than the specified time set by the [SEARCH:SEARCH<x>:TRIGger:A:LOGic:PATtern:WHEn:MORELimit](#) command.

**SEARCH:SEARCH<x>:TRIGger:A:LOGic:PATtern:WHEn:LESSLimit**

This command sets or returns the maximum time that the selected pattern may be true and still generate an A logic pattern search to place a mark.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:LOGic:PATtern:WHEn:LESSLimit  
<NR3>  
SEARCH:SEARCH<x>:TRIGger:A:LOGic:PATtern:WHEn:LESSLimit?

**Arguments** <NR3> specifies the maximum amount of time to hold the pattern true.

**SEARCH:SEARCH<x>:TRIGger:A:LOGic:PATtern:WHEn:MORELimit**

This command sets or returns the minimum time that the selected pattern may be true and still generate an A logic pattern search to place a mark.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:LOGic:PATtern:WHEn:MORELimit  
<NR3>  
SEARCH:SEARCH<x>:TRIGger:A:LOGic:PATtern:WHEn:MORELimit?

**Arguments** <NR3> specifies the minimum amount of time to hold the pattern true.

**SEARCH:SEARCH<x>:TRIGger:A:LOGic:THReshold:CH<x>**

This command sets or returns the channel threshold level for an edge search to determine where to place a mark.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:LOGic:THReshold:CH<x>  
{<NR3>|ECL|TTL}  
SEARCH:SEARCH<x>:TRIGger:A:LOGic:THReshold:CH<x>?

**Arguments** <NR3> specifies the trigger level, in volts.

ECL specifies a -1.3 V trigger level.

TTL specifies a 1.4 V trigger level.

## SEARCH:SEARCH<x>:TRIGger:A:LOGic:THReshold:MATH

This command sets or returns the math waveform threshold level for edge search to determine where to place a mark

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:LOGic:THReshold:MATH {ECL|TTL}  
SEARCH:SEARCH<x>:TRIGger:A:LOGic:THReshold:MATH?

**Arguments** ECL specifies a –1.3 V trigger level.  
TTL specifies a 1.4 V trigger level.

## SEARCH:SEARCH<x>:TRIGger:A:LOGic:THReshold:REF<x>

This command sets or returns the reference waveform threshold level for edge search to determine where to place a mark

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:LOGic:THReshold:REF<x> {ECL|TTL}  
SEARCH:SEARCH<x>:TRIGger:A:LOGic:THReshold:REF<x>?

**Arguments** ECL specifies a –1.3 V trigger level.  
TTL specifies a 1.4 V trigger level.

## SEARCH:SEARCH<x>:TRIGger:A:LOWerthreshold:CH<x>

This command sets or returns the channel waveform lower threshold level for a transition search to determine where to place a mark.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:LOWerthreshold:CH<x> {ECL|TTL}  
SEARCH:SEARCH<x>:TRIGger:A:LOWerthreshold:CH<x>?

**Arguments** ECL specifies a –1.3 V trigger level.  
TTL specifies a 1.4 V trigger level.

## **SEARCH:SEARCH<x>:TRIGger:A:LOWerthreshold:MATH**

This command sets or returns the math waveform lower threshold level for a transition search to determine where to place a mark.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:LOWerthreshold:MATH {ECL|TTL}  
SEARCH:SEARCH<x>:TRIGger:A:LOWerthreshold:MATH?

**Arguments** ECL specifies a –1.3 V trigger level.  
TTL specifies a 1.4 V trigger level.

## **SEARCH:SEARCH<x>:TRIGger:A:LOWerthreshold:REF<x>**

This command sets or returns the reference waveform lower threshold level for a transition search to determine where to place a mark.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:LOWerthreshold:REF<x> {ECL|TTL}  
SEARCH:SEARCH<x>:TRIGger:A:LOWerthreshold:REF<x>?

**Arguments** ECL specifies a –1.3 V trigger level.  
TTL specifies a 1.4 V trigger level.

## **SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:POLarity**

This command sets or returns the polarity for a pulse search to determine where to place a mark.

**Group** Search

**Syntax**     SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:POLarity  
                   {NEGative|POSitive}  
                   SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:POLarity?

**Arguments**     POSITIVE specifies that the instrument will only place a mark when the polarity of the pulse is positive.

                  NEGative specifies that the instrument will only place a mark when the polarity of the pulse is negative.

## SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:SOURce

This command sets or returns the source waveform for a pulse search to determine where to place a mark.

**Group**     Search

**Syntax**     SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:SOURce  
                   {CH1|CH2|CH3|CH4|MATH}  
                   SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:SOURce?

**Arguments**     CH<x> specifies one input channel as the edge source, where <x> = 1, 2, 3 or 4.

                  MATH specifies the math waveform as the search source.

## SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:WHEn

This command sets or returns the condition for generating a pulse width search to determine where to place a mark.

**Group**     Search

**Syntax**     SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:WHEn  
                   {LESSthan|MOREthan|EQua1|UNEQua1}  
                   SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:WHEn?

**Arguments**     LESSThan argument sets the instrument to place a mark if the pulse width is less than the time set by the [SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:WIDth](#) command.



MOREThan argument sets the instrument to place a mark if the pulse width is true longer than the specified time set by the [SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:WIDTH](#) command.

EQUa1 argument sets the instrument to place a mark if the pulse width is equal to the time set by the [SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:WIDTH](#) command within a tolerance of  $\pm 5\%$ .

UNEQUa1 argument sets the instrument to place a mark if the pulse width is unequal to the time the time set by the [SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:WIDTH](#) command within a tolerance of  $\pm 5\%$ .

## SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:WIDTH

This command sets or returns the pulse width setting for a pulse width search to determine where to place a mark.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:WIDTH <NR3>  
SEARCH:SEARCH<x>:TRIGger:A:PULSEwidth:WIDTH?

**Arguments** <NR3> is the pulse width.

## SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLarity

This command sets or returns the polarity setting for a runt search to determine where to place a mark.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLarity  
{EITher|NEGative|POSitive}  
SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLarity?

**Arguments** POSitive indicates that the rising edge crosses the low threshold and the falling edge re-crosses the low threshold without either edge ever crossing the high threshold.

NEGative indicates that the falling edge crosses the high threshold and the rising edge re-crosses the high threshold without either edge ever crossing the low threshold.

**EITHer** finds on a runt of either polarity.

## SEARCH:SEARCH<x>:TRIGger:A:RUNT:SOURce

This command sets or returns the source setting for a runt search to determine where to place a mark.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:RUNT:SOURce  
{CH1|CH2|CH3|CH4|MATH}  
SEARCH:SEARCH<x>:TRIGger:A:RUNT:SOURce?

**Arguments** CH<x> specifies one input channel as the edge source, where <x> = 1, 2, 3 or 4.  
MATH specifies the math waveform as the search source.

## SEARCH:SEARCH<x>:TRIGger:A:RUNT:WHEn

This command sets or returns the condition setting for a runt search to determine where to place a mark.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:RUNT:WHEn  
{LESSthan|MOREthan|EQua1|UNEQua1|OCCURS}  
SEARCH:SEARCH<x>:TRIGger:A:RUNT:WHEn?

**Arguments** OCCURS argument specifies a trigger event if a runt of any detectable width occurs.  
  
LESSthan argument sets the instrument to trigger if the a runt pulse is detected with width less than the time set by the [SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDth](#) command.  
  
MOREthan argument sets the instrument to trigger if the a runt pulse is detected with width more than the time set by the [SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDth](#) command.  
  
EQua1 argument sets the instrument to trigger when the pattern is true for a time period equal to the time period specified in [SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDth](#) within a  $\pm 5\%$  tolerance.

NOTEQUAL argument sets the instrument to trigger when the pattern is true for a time period greater than or less than (but not equal) the time period specified in [SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDTH](#) within a  $\pm 5\%$  tolerance.

## SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDTH

This command sets or returns the width setting for a runt search to determine where to place a mark.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDTH <NR3>  
SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDTH?

**Arguments** <NR3> specifies the minimum width, in seconds.

## SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCK:EDGE

This command sets or returns the clock slope setting for a setup/hold search to determine where to place a mark.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCK:EDGE {FALL|RISe}  
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCK:EDGE?

**Arguments** FALL specifies polarity as the clock falling edge.  
RISe specifies polarity as the clock rising edge.

## SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCK:SOURce

This command sets or returns the clock source setting for an setup/hold search to determine where to place a mark.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCK:SOURce  
{CH1|CH2|CH3|CH4|MATH}

SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCK:SOURce?

**Arguments** CH<x> specifies one input channel as the edge source, where <x> = 1, 2, 3 or 4.  
MATH specifies the math waveform as the search source.

## SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCK:THReshold

This command sets or returns the clock threshold setting for an setup/hold search to determine where to place a mark.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCK:THReshold  
{<NR3>|ECL|TTL}  
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCK:THReshold?

**Arguments** ECL specifies a preset ECL high level of -1.3 V.  
TTL specifies a preset TTL high level of 1.4 V.  
<NR3> is the clock level, in volts.

## SEARCH:SEARCH<x>:TRIGger:A:SETHold:DATA:SOURce

This command sets or returns the data source setting for an setup/hold search to determine where to place a mark.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:SETHold:DATA:SOURce  
{CH1|CH2|CH3|CH4|MATH}  
SEARCH:SEARCH<x>:TRIGger:A:SETHold:DATA:SOURce?

**Arguments** CH<x> specifies one input channel as the edge source, where <x> = 1, 2, 3 or 4.  
MATH specifies the math waveform as the search source.

## SEARCH:SEARCH<x>:TRIGger:A:SETHold:DATA:THReshold

This command sets or returns the data threshold setting for an setup/hold search to determine where to place a mark.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:SETHold:DATA:THReshold  
{<NR3>|ECL|TTL}  
SEARCH:SEARCH<x>:TRIGger:A:SETHold:DATA:THReshold?

**Arguments** ECL specifies a preset ECL high level of -1.3 V.  
TTL specifies a preset TTL high level of 1.4 V.  
<NR3> is the clock level, in volts.

## SEARCH:SEARCH<x>:TRIGger:A:SETHold:HOLDTime

This command sets or returns the hold time setting for an setup/hold search to determine where to place a mark.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:SETHold:HOLDTime <NR3>  
SEARCH:SEARCH<x>:TRIGger:A:SETHold:HOLDTime?

**Arguments** <NR3> specifies the hold time setting in seconds. Positive values for hold time occur after the clock edge. Negative values occur before the clock edge.

## SEARCH:SEARCH<x>:TRIGger:A:SETHold:SETTime

This command sets or returns the setup time setting for an setup/hold search to determine where to place a mark.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:SETHold:SETTime <NR3>  
SEARCH:SEARCH<x>:TRIGger:A:SETHold:SETTime?

**Arguments** <NR3> specifies the setup time for setup and hold violation triggering.

## SEARCH:SEARCH<x>:TRIGger:A{:TRANSition|:RISEFall}:DELTatime

This command sets or returns the transition time setting for an transition search to determine where to place a mark.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A{:TRANSition|:RISEFall}:DELTatime  
<NR3>  
SEARCH:SEARCH<x>:TRIGger:A{:TRANSition|:RISEFall}:DELTatime?

**Arguments** <NR3> specifies the transition time, in seconds.

## SEARCH:SEARCH<x>:TRIGger:A{:TRANSition|:RISEFall}:POLarity

This command sets or returns the polarity setting for an transition search to determine where to place a mark.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A{:TRANSition|:RISEFall}:POLarity  
{EITHer|NEGative|POSitive}  
SEARCH:SEARCH<x>:TRIGger:A{:TRANSition|:RISEFall}:POLarity?

**Arguments** POSitive indicates that a pulse edge must traverse from the lower (most negative) to higher (most positive) level for transition triggering to occur.

NEGative indicates that a pulse edge must traverse from the upper (most positive) to lower (most negative) level for transition triggering to occur.

EITHer indicates either positive or negative polarity.

## SEARCH:SEARCH<x>:TRIGger:A{:TRANSition|:RISEFall}:SOUrce

This command sets or returns the source setting for an transition search to determine where to place a mark.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A{:TRANSition|:RISEFall}:SOURCE  
{CH1|CH2|CH3|CH4|MATH}  
SEARCH:SEARCH<x>:TRIGger:A{:TRANSition|:RISEFall}:SOURCE?

**Arguments** CH<x> specifies one input channel as the edge source, where <x> = 1, 2, 3 or 4.  
MATH specifies the math waveform as the search source.

## SEARCH:SEARCH<x>:TRIGger:A{:TRANSition|:RISEFall}:WHEN

This command sets or returns the condition setting for an transition search to determine where to place a mark.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A{:TRANSition|:RISEFall}:WHEN  
{SLOWer|FASTer|EQua1|UNEQua1}  
SEARCH:SEARCH<x>:TRIGger:A{:TRANSition|:RISEFall}:WHEN?

**Arguments** FASTer sets the trigger to occur when the transitioning signal is faster than the set volts/second rate.  
  
SLOWer sets the trigger to occur when the transitioning signal is slower than the set volts/second rate.  
  
EQua1 sets the trigger to occur when the transitioning signal is equal to the set volts/second rate within a  $\pm 5\%$  tolerance.  
  
UNEQua1 sets the trigger to occur when the transitioning signal is not equal to the set volts/second rate  $\pm 5\%$ .

## SEARCH:SEARCH<x>:TRIGger:A:TYPE

This command sets or returns the trigger type setting for a search to determine where to place a mark.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:TYPE  
{EDGE|SETHold|PULSEwidth|RUNt|TRANSition}  
SEARCH:SEARCH<x>:TRIGger:A:TYPE?

<b>Arguments</b>	<p><b>RUNt</b> triggers when a pulse crosses the first preset voltage threshold but does not cross the second preset threshold before recrossing the first. The thresholds are set with the <code>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOWerthreshold:CH&lt;x&gt;</code> and <code>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:UPPerthreshold:CH&lt;x&gt;</code> commands.</p> <p><b>PULSEWidth</b> triggers when a pulse is found that has the specified polarity and is either inside or outside the limits as specified by <code>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOGic:PATtern:WHEn:LESSLimit</code> and <code>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:LOGic:PATtern:WHEn:MORELimit</code>. The polarity is selected using the <code>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:RUNt:POLarity</code> command.</p> <p><b>TRANSition</b> triggers when a pulse crosses both thresholds in the same direction as the specified polarity and the transition time between the two threshold crossings is greater or less than the specified time delta.</p>
------------------	--

## SEARCH:SEARCH<x>:TRIGger:A:UPPerthreshold:CH<x>

This command sets or returns the channel waveform upper threshold level for a transition search to determine where to place a mark.

<b>Group</b>	Search
<b>Syntax</b>	<pre>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:UPPerthreshold:CH&lt;x&gt; {ECL TTL} SEARCH:SEARCH&lt;x&gt;:TRIGger:A:UPPerthreshold:CH&lt;x&gt;?</pre>
<b>Arguments</b>	<p>ECL specifies a preset ECL high level of -1.3 V.</p> <p>TTL specifies a preset TTL high level of 1.4 V.</p>

## SEARCH:SEARCH<x>:TRIGger:A:UPPerthreshold:MATH

This command sets or returns the math waveform upper threshold level for a transition search to determine where to place a mark.

<b>Group</b>	Search
<b>Syntax</b>	<pre>SEARCH:SEARCH&lt;x&gt;:TRIGger:A:UPPerthreshold:MATH {ECL TTL} SEARCH:SEARCH&lt;x&gt;:TRIGger:A:UPPerthreshold:MATH?</pre>
<b>Arguments</b>	<p>ECL specifies a preset ECL high level of -1.3 V.</p> <p>TTL specifies a preset TTL high level of 1.4 V.</p>



**SEARCH:SEARCH<x>:TRIGger:A:UPPerthreshold:REF<x>**

This command sets or returns the reference waveform upper threshold level for a transition search to determine where to place a mark.

**Group** Search

**Syntax** SEARCH:SEARCH<x>:TRIGger:A:UPPerthreshold:REF<x> {ECL|TTL}  
SEARCH:SEARCH<x>:TRIGger:A:UPPerthreshold:REF<x>?

**Arguments** ECL specifies a preset ECL high level of -1.3 V.  
TTL specifies a preset TTL high level of 1.4 V.

**SElect? (Query Only)**

This query-only command returns the selected waveform that is affected by the front-panel controls and the display status (on or off) of all waveforms. This query command is equivalent to selecting Measurement Setup from the Measure menu and viewing the Source waveform setting.

**Group** Vertical

**Syntax** SElect?

**Examples** SELECT? might return the waveform and status information as :SELECT:CH1 1;CH2 0;CH3 0;CH4 0;MATH1 0; MATH2 0;MATH3 0;MATH4 0;REF1 0;REF2 0; REF3 0;REF4 0;CONTROL CH1

**SElect:BUS<x>**

This command turns on the display of the specified Bus waveform and also resets the acquisition. The query returns whether the specified Bus waveform is on or off but does not indicate whether it is the selected waveform. The value of <x> can be 1 or 2.

**Group** Vertical

**Syntax** SElect:BUS<x> {<NR1>|OFF|ON}  
SElect:BUS<x>?

<b>Arguments</b>	<p>ON turns on the display of the specified waveform. This waveform also becomes the selected waveform.</p> <p>OFF turns off the display of the specified waveform.</p> <p>&lt;NR1&gt; = 0 turns off the display of the specified waveform; any other value turns on the display of the specified waveform.</p>
<b>Examples</b>	<p>SELECT:BUS1 ON turns the Bus 1 waveform display on, and selects it.</p> <p>SELECT:BUS1? might return :SELECT:BUS1 1 indicating that Bus 1 waveform is being displayed.</p>

## SElect:CH<x>

This command turns on the display of a specified waveform and also resets the acquisition. The query returns whether the channel is on or off but does not indicate whether it is the selected waveform.

<b>Group</b>	Vertical
<b>Syntax</b>	<p>SElect:CH&lt;x&gt; {ON OFF &lt;NR1&gt;}</p> <p>SElect:CH&lt;x&gt;?</p>
<b>Arguments</b>	<p>ON turns on the display of the specified waveform. This waveform also becomes the selected waveform.</p> <p>OFF turns off the display of the specified waveform.</p> <p>&lt;NR1&gt; = 0 turns off the display of the specified waveform; any other value turns on the display of the specified waveform.</p>
<b>Examples</b>	<p>SELECT:CH2 ON turns the channel 2 waveform display on, and selects channel 2.</p> <p>SELECT:CH1? might return :SELECT:CH1 1 indicating that channel 1 is being displayed.</p>

## SElect:CONTROL <wfm>

This command sets or returns the waveform that is selected as the implied recipient of channel-related commands that support legacy-style programs. The command form also performs the equivalent of a [SElect:CH<x> ON](#) command. This command is equivalent to selecting Measurement Setup from the Measure menu and either viewing or setting the Source waveform.

<b>Group</b>	Vertical
<b>Syntax</b>	<code>SElect:CONTROL {CH&lt;x&gt; MATH}</code> <code>SElect:CONTROL?</code>
<b>Arguments</b>	<p>CH&lt;x&gt; selects the specified channel waveform as the waveform that is affected by the front-panel controls. The x variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.</p> <p>MATH selects the math waveform as the waveform that is affected by the front-panel controls.</p>
<b>Examples</b>	<p><code>SELECT:CONTROL CH2</code> resets acquisition displays on channel 2, and causes the selected waveform to be the implied object of waveform commands.</p> <p><code>SELECT:CONTROL?</code> might return <code>:SELECT:CONTROL MATH</code> indicating that math is the implied object of waveform commands.</p>

## SElect:MATH[1]

This command turns on the display of the math waveform and also resets the acquisition. The query returns whether the math waveform is on or off but does not indicate whether it is the selected waveform.

<b>Group</b>	Vertical
<b>Syntax</b>	<code>SElect:MATH[1] {ON OFF &lt;NR1&gt;}</code> <code>SElect:MATH[1]?</code>
<b>Arguments</b>	<p>ON turns on the display of the specified waveform. This waveform also becomes the selected waveform.</p> <p>OFF turns off the display of the specified waveform.</p> <p>&lt;NR1&gt; = 0 turns off the display of the specified waveform; any other value turns on the display of the specified waveform.</p>
<b>Examples</b>	<p><code>SELECT:MATH ON</code> turns the math waveform display on, and selects it.</p> <p><code>SELECT:MATH?</code> might return <code>:SELECT:MATH 1</code> indicating that the math waveform is being displayed.</p>

## SElect:REF<x>

This command turns on the display of a specified reference waveform and also resets the acquisition. The query returns whether the channel is on or off but does not indicate whether it is the selected waveform.

**Group** Vertical

**Syntax** SElect:REF<x> {ON|OFF|<NR1>}  
SElect:REF<x>?

**Arguments** ON turns on the display of the specified waveform. This waveform also becomes the selected waveform.

OFF turns off the display of the specified waveform.

<NR1> = 0 turns off the display of the specified waveform; any other value turns on the display of the specified waveform.

**Examples** SELECT:REF2 ON turns the channel 2 waveform display on, and selects reference waveform 2.

SELECT:REF3? might return :SELECT:REF3 1 indicating that reference waveform 3 is being displayed.

## SET? (Query Only)

This query-only command returns the commands that list the instrument settings, except for configuration information for the calibration values. You can use these commands to return the instrument to the state it was in when you made the SET? query. The SET? query always returns command headers, regardless of the setting of the [HEADer](#) command. This is because the returned commands are intended to be sent back to the instrument as a command string. The [VERBose](#) command can still be used to specify whether the returned headers should be abbreviated or full-length.

This command is identical to the [\\*LRN?](#) command.

**Group** Miscellaneous

**Syntax** SET?

**Related Commands** [HEADer](#), [\\*LRN?](#), [VERBose](#)

**Examples** SET? might return the following response: :ACQUIRE:STOPAFTER  
 RUNSTOP;STATE 1;MODE SAMPLE;NUMENV 10;NUMAVG 16;REPET  
 1;:FASTACQ:STATE 0;:APPLICATION:GPKNOB1:ACTIVE  
 0;:APPLICATION:GPKNOB2:ACTIVE 0;:APPLICATION:WINDOW:HEIGHT  
 236;WIDTH 640;:APPLICATION:SCOPEAPP:STATE  
 RUNNING;WINDOW FULLSCREEN;:APPLICATION:EXTAPP:STATE  
 NOTRUNNING;:AUXOUT:SOURCE ATRIGGER;EDGE  
 FALLING;:CMD BATCH 1;:HEADER 1;:LOCK NONE;:ROSC:SOURCE  
 INTERNAL;:VERBOSE 1;:ALIAS:STATE 0;:DISPLAY:CLOCK  
 1;COLOR:PALETTE NORMAL;MATHCOLOR DEFAULT;REFCOLOR  
 DEFAULT;:DISPLAY:FILTER SINX;FORMAT YT;GRATICULE  
 FULL;INTENSITY:WAVEFORM 75.0000;AUTOBRIGHT 1;SCREENSAVER  
 1;SCREENSAVERDELAY 28800;:DISPLAY:PERSISTENCE  
 OFF;STYLE VECTORS;TRIGBAR SHORT;TRIGT 1;VARPERSIST  
 500.0000E-3;:HARDCOPY:FILENAME "untitled";PORT  
 FILE;:DIAG:LEVEL SUBSYS;:SAVE:WAVEFORM:FILEFORMAT  
 INTERNAL;:TRIGGER:A:MODE AUTO;TYPE  
 EDGE;LEVEL 0.0000;HOLDOFF:BY DEFAULT;TIME  
 1.5000E-6;:TRIGGER:A:EDGE:SOURCE CH1;COUPLING DC;SLOPE RISE;

## \*SRE

The \*SRE (Service Request Enable) command sets and queries the bits in the Service Request Enable Register. For more information, refer to Registers.

**Group** Status and Error

**Syntax** \*SRE <NR1>  
 \*SRE?

**Related Commands** [\\*CLS](#), [DESE](#), [\\*ESE](#), [\\*ESR?](#), [EVENT?](#), [EVMsg?](#), [FACtory](#), [\\*STB?](#)

**Arguments** <NR1> is a value in the range from 0 through 255. The binary bits of the SRER are set according to this value. Using an out-of-range value causes an execution error. The power-on default for SRER is 0 if \*PSC is 1. If \*PSC is 0, the SRER maintains the previous power cycle value through the current power cycle.

**Examples** \*SRE 48 sets the bits in the SRER to binary 00110000.  
 \*SRE might return 32, showing that the bits in the SRER have the binary value of 00100000.

## STArtuptime? (Query Only)

Returns the time that the oscilloscope was last powered on.

**Group** Miscellaneous

**Syntax** STArtuptime?

## \*STB? (Query Only)

The \*STB? (Read Status Byte) query returns the contents of the Status Byte Register (SBR) using the Master Summary Status (MSS) bit. For more information, refer to Registers.

**Group** Status and Error

**Syntax** \*STB?

**Related Commands** [\\*CLS](#), [DESE](#), [\\*ESE](#), [\\*ESR?](#), [EVENT?](#), [EVMsg?](#), [FACtory](#), [\\*SRE](#)

**Returns** <NR1>

**Examples** \*STB? might return 96, showing that the SBR contains the binary value 01100000.

## TEKSecure (No Query Form)

This command initializes both waveform and setup memories, overwriting any previously stored data.

TEKSecure deletes all four waveform reference memory slots on the hard drive, if they exist, and puts all setups in the factory-initialized state. External setups that are stored on the hard drive are not affected.

**Group** Miscellaneous

**Syntax** TEKSecure

**Examples**     TEKSECURE initializes both waveform and setup memories.

## TIME

This command sets or returns the time that the instrument displays. This command is equivalent to selecting Set Time & Date from the Utilities menu and then setting the fields in the Time group box.

**Group**     Miscellaneous

**Syntax**     TIME <QString>  
TIME?

**Related Commands**     [DATE](#)

**Arguments**     <QString> is a time in the form "hh:mm:ss" where hh refers to a two-digit hour number, mm refers to a two-digit minute number from 01 to 60, and ss refers to a two-digit second number from 01 to 60.

**Examples**     TIME "14:00:00" sets the time to exactly 2:00 p.m.  
  
TIME? might return :TIME "14:05:17" indicating the current time is set to 2:05 p.m. and 17 seconds.

## TOTaluptime? (Query Only)

This command returns the total number of hours that the oscilloscope has been turned on since the nonvolatile memory was last programmed (usually since the initial manufacturing process).

**Group**     Miscellaneous

**Syntax**     TOTaluptime?

## \*TRG (No Query Form)

This command (no query form) performs the group execute trigger on commands defined by \*DDT.

<b>Group</b>	Miscellaneous
<b>Syntax</b>	*TRG
<b>Related Commands</b>	<a href="#">*DDT</a>
<b>Examples</b>	*TRG immediately executes all commands that have been defined by *DDT.

## TRIGger (No Query Form)

This command forces a trigger event to occur.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger FORCE
<b>Arguments</b>	FORCE creates a trigger event. If TRIGger:STATE is set to READy, the acquisition will complete. Otherwise, this command will be ignored. This is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu and then clicking Force Trigger.
<b>Examples</b>	TRIGGER FORCE forces a trigger event to occur.

## TRIGger:A

This command sets the A trigger level automatically to 50% of the range of the minimum and maximum values of the trigger input signal. The query returns current A trigger parameters. The trigger level is the voltage threshold through which the trigger source signal must pass to generate a trigger event. This command is equivalent to pushing the LEVEL knob on the front panel.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:A SETLeve1 TRIGger:A?
<b>Related Commands</b>	<a href="#">TRIGger:A:EDGE?</a> , <a href="#">TRIGger:A:LOGic?</a> , <a href="#">TRIGger:A:PULse?</a>



**Arguments** SETLevel sets the A trigger level to 50% of the range of the minimum and maximum values of the trigger input signal.

**Examples** TRIGGER:A SETLEVEL sets the A trigger level to 50% of the range of the minimum and maximum values of the trigger input signal.

TRIGGER:A? might return the following A trigger parameters:

```
:TRIGGER:A:MODE AUTO;TYPE EDGE;LEVEL 0.0000;HOLDOFF:BY
DEFAULT;TIME 1.5000E-6;;:TRIGGER:A:EDGE:SOURCE
CH1;COUPLING DC;SLOPE RISE;;:TRIGGER:A:LOGIC:CLASS
PATTERN;FUNCTION AND;THRESHOLD:CH1 1.2000;CH2
1.2000;CH3 1.2000;CH4 1.2000;;:TRIGGER:A:LOGIC:INPUT:CH1
HIGH;CH2 X;CH3 X;;:TRIGGER:A:LOGIC:PATTERN:INPUT:CH4
X;;:TRIGGER:A:LOGIC:PATTERN:WHEN
TRUE;WHEN:LESSLIMIT 5.0000E-9;MORELIMIT
5.0000E-9;;:TRIGGER:A:LOGIC:SETHOLD:CLOCK:EDGE RISE;THRESHOLD
1.2000;SOURCE CH2;;:TRIGGER:A:LOGIC:SETHOLD:DATA:THRESHOLD
1.2000;SOURCE CH1;;:TRIGGER:A:LOGIC:SETHOLD:HOLDTIME
2.0000E-9;SETTIME 3.0000E-9;;:TRIGGER:A:LOGIC:STATE:INPUT:CH4
RISE;;:TRIGGER:A:LOGIC:STATE:WHEN
TRUE;;:TRIGGER:A:PULSE:CLASS GLITCH;SOURCE
CH1;GLITCH:WIDTH 2.0000E-9;TRIGIF ACCEPT;POLARITY
POSITIVE;;:TRIGGER:A:PULSE:WINDOW:TYPE INSIDE;WHEN
OCCURS;WIDTH 2.0000E-9;LOGIC:INPUT:CH1 HIGH;CH2 HIGH;CH3
HIGH;CH4 HIGH;;:TRIGGER:A:PULSE:WINDOW:LOGIC:THRESHOLD:CH1
1.2000;CH2 1.2000;CH3 1.2000;CH4
1.2000;;:TRIGGER:A:PULSE:WINDOW:THRESHOLD:HIGH 1.2000;LOW
800.0000E-3;;:TRIGGER:A:PULSE:RUNT:LOGIC:INPUT:CH1
HIGH;CH2 HIGH;CH3 HIGH;CH4
HIGH;;:TRIGGER:A:PULSE:RUNT:LOGIC:THRESHOLD:CH1 1.2000;CH2
1.2000;CH3 1.2000;CH4 1.2000;;:TRIGGER:A:PULSE:RUNT:POLARITY
POSITIVE;THRESHOLD:HIGH 1.2000;LOW
800.0000E-3;;:TRIGGER:A:PULSE:RUNT:WHEN OCCURS;WIDTH
2.0000E-9;;:TRIGGER:A:PULSE:TRANSITION:DELTATIME
2.0000E-9;POLARITY POSITIVE;THRESHOLD:HIGH 1.2000;LOW
800.0000E-3;;:TRIGGER:A:PULSE:TRANSITION:WHEN
FASTER THAN;;:TRIGGER:A:PULSE:WIDTH:LOWLIMIT
2.0000E-9;HIGHLIMIT 2.0000E-9;WHEN WITHIN;POLARITY
POSITIVE;;:TRIGGER:A:PULSE:TIMEOUT:POLARITY
STAYSHIGH;TIME 2.0000E-9;;:TRIGGER:A:VIDEO:CUSTOM:FORMAT
INTERLACED;SCAN RATE1;;:TRIGGER:A:VIDEO:FIELD
ALLFIELDS;HOLDOFF:FIELD 0.0000;;:TRIGGER:A:VIDEO:LINE
1;POLARITY NORMAL;SCAN RATE1;SOURCE CH1;STANDARD
NTSC;;:TRIGGER:A:COMMUNICATION:STANDARD CUSTOM;CODE
NRZ;BITRATE 1544000;AMI:PULSEFORM PLUSONE;THRESHOLD:HIGH
0.0000;LOW 0.0000;;:TRIGGER:A:COMMUNICATION:CLOCK:POLARITY
```

RISE; :TRIGGER:A:COMMUNICATION:CMI:PULSEFORM  
 PLUSONE; :TRIGGER:A:COMMUNICATION:SOURCE CH1;SOURCE:TYPE DATA

## TRIGger:A:BUS

This command sets or queries the serial trigger type: I2C, CAN or SPI.

<b>Conditions</b>	This command requires a DPO4AUTO or DPO4EMBD application module.
<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:A:BUS {I2C SPI CAN} TRIGger:A:BUS?
<b>Arguments</b>	I2C specifies the Inter-IC bus.  SPI specifies the Serial Peripheral Interface bus.  CAN specifies the Controller Area Network bus.

## TRIGger:A:BUS:B<x>:CAN:CONDition

This command sets or returns the CAN condition.

<b>Conditions</b>	This command requires a DPO4AUTO application module.
<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:A:BUS:B<x>:CAN:CONDition {SOF FRAMEtype Identifier DATA IDANDDATA EOF ACKMISS} TRIGger:A:BUS:B<x>:CAN:CONDition?
<b>Arguments</b>	SOF enables triggering on the start of frame.  FRAME enables triggering on the type of frame.  IDENTIFIER enables triggering on a matching identifier.  DATA enables triggering on matching data.  IDENTANDDATA enables triggering on a matching identifier and matching data.  EOF enables triggering on the end of frame.

ACKMISS enables triggering on a missing acknowledge.

ERROR enables triggering on an error within a frame.

**Examples**     `TRIGGER:A:BUS:B1:CAN:CONDITION?` might return `:TRIGGER:A:BUS:B1:CAN:CONDITION EOF` indication an end of file condition.

`TRIGGER:A:BUS:B1:CAN:CONDITION DATA` enables triggering on matching CAN data.

## **TRIGger:A:BUS:B<x>:CAN:DATA:DIRection**

This command sets or queries the CAN trigger condition to be valid on a READ, WRITE, or either. This only applies if the trigger condition is ID.

**Conditions**     This command requires a DPO4AUTO application module.

**Group**     Trigger

**Syntax**     `TRIGger:A:BUS:B<x>:CAN:DATA:DIRection {READ|WRITE|NOCARE}`  
`TRIGger:A:BUS:B<x>:CAN:DATA:DIRection?`

**Arguments**     READ sets the CAN data direction to READ.

                    WRITEsets the CAN data direction to WRITE..

                    NOCAREsets the CAN data direction to either.

**Examples**     `TRIGGER:A:BUS:B1:CAN:DATA:DIRECTION WRITE` sets the CAN data direction to Write.

## **TRIGger:A:BUS:B<x>:CAN:DATA:QUALifier**

This command sets or returns the CAN data qualifier. This only applies if the trigger condition is IDANDDATA or DATA.

**Conditions**     This command requires a DPO4AUTO application module.

**Group**     Trigger

**Syntax**     TRIGger:A:BUS:B<x>:CAN:DATA:QUALifier  
                 {LESSThan|MOREThan|EQUal|UNEQUal}  
                 TRIGger:A:BUS:B<x>:CAN:DATA:QUALifier?

## TRIGger:A:BUS:B<x>:CAN:DATA:SIZE

This command sets or returns the length of the data string in bytes to be used for CAN trigger, if trigger condition is DATA or IDANDDATA.

**Conditions**     This command requires a DPO4AUTO application module.

**Group**           Trigger

**Syntax**           TRIGger:A:BUS:B<x>:CAN:DATA:SIZE <NR3>  
                 TRIGger:A:BUS:B<x>:CAN:DATA:SIZE?

**Arguments**       <NR3> is the length of the data string in bytes.

## TRIGger:A:BUS:B<x>:CAN:DATA:VALue

This command sets or returns the binary data string to be used for CAN trigger if trigger condition is ID or IDANDDATA.

**Conditions**       This command requires a DPO4AUTO application module.

**Group**           Trigger

**Syntax**           TRIGger:A:BUS:B<x>:CAN:DATA:VALue <NR3>  
                 TRIGger:A:BUS:B<x>:CAN:DATA:VALue?

**Arguments**       <NR3> is the CAN data value.

**Examples**         TRIGGER:A:BUS:B1:CAN:DATA:VALUE 1011 sets the CAN data value to 1011.

## TRIGger:A:BUS:B<x>:CAN:FRAMEtype

This command sets or queries the CAN trigger frame type.

<b>Conditions</b>	This command requires a DPO4AUTO application module.
<b>Group</b>	Trigger
<b>Syntax</b>	<code>TRIGger:A:BUS:B&lt;x&gt;:CAN:FRAMetype</code> <code>{DATA REMOte ERRor OVERload}</code> <code>TRIGger:A:BUS:B&lt;x&gt;:CAN:FRAMetype?</code>
<b>Arguments</b>	DATA specifies a data frame type.  REMOTE specifies a remote frame type.  ERROR specifies an error frame type.  OVERLOAD specifies an overload frame type.
<b>Examples</b>	<code>TRIGGER:A:BUS:B1:CAN:FRAMETYPE DATA</code> sets the CAN trigger frame type to DATA.

## **TRIGger:A:BUS:B<x>:CAN{:IDentifier|:ADDRess}:MODE**

This command sets or returns the CAN addressing mode to standard or extended format.

<b>Conditions</b>	This command requires a DPO4AUTO application module.
<b>Group</b>	Trigger
<b>Syntax</b>	<code>TRIGger:A:BUS:B&lt;x&gt;:CAN{:IDentifier :ADDRess}:MODE</code> <code>{STandard EXTended}</code> <code>TRIGger:A:BUS:B&lt;x&gt;:CAN{:IDentifier :ADDRess}:MODE?</code>
<b>Arguments</b>	STANDARD specifies the standard addressing mode.  EXTENDED specifies the extended addressing mode.

## **TRIGger:A:BUS:B<x>:CAN{:IDentifier|:ADDRess}:VALue**

This command sets or queries the binary address string used for the CAN trigger if the trigger condition is ID or IDANDDATA.

<b>Conditions</b>	This command requires a DPO4AUTO application module.
<b>Group</b>	Trigger
<b>Syntax</b>	<pre>TRIGger:A:BUS:B&lt;x&gt;:CAN{:IDentifier :ADDRESS}:VALUE &lt;NR3&gt; TRIGger:A:BUS:B&lt;x&gt;:CAN{:IDentifier :ADDRESS}:VALUE?</pre>
<b>Arguments</b>	<NR3> is up to 29 bits specifying the CAN identifier value.
<b>Examples</b>	TRIGGER:A:BUS:B1:CAN:IDENTIFIER:VALUE 1011 sets the CAN trigger identifier value to 1011.

## TRIGger:A:BUS:B<x>:I2C:ADDRESS:MODE

This command sets or queries the I2C address mode to 7 or 10-bit.

<b>Conditions</b>	This command requires a DPO4EMBD application module.
<b>Group</b>	Trigger
<b>Syntax</b>	<pre>TRIGger:A:BUS:B&lt;x&gt;:I2C:ADDRESS:MODE {ADDR7 ADDR10} TRIGger:A:BUS:B&lt;x&gt;:I2C:ADDRESS:MODE?</pre>
<b>Arguments</b>	<p>ADDR7 specifies the 7-bit I2C address mode.</p> <p>ADDR10 specifies the 10-bit I2C address mode.</p>
<b>Examples</b>	TRIGGER:A:BUS:B1:I2C:ADDRESS:MODE ADDR10 sets the I2C address mode to 10-bit.

## TRIGger:A:BUS:B<x>:I2C:ADDRESS:TYPE

This command sets or returns the I2C address type to I2C special addresses (general call, HS-mode etc.), or to a user-specified address.

<b>Conditions</b>	This command requires a DPO4EMBD application module.
<b>Group</b>	Trigger

**Syntax**     `TRIGger:A:BUS:B<x>:I2C:ADDRESS:TYPE`  
                  `{Generalcall|STARTbyte|HSmode|EEPROM|USER}`  
                  `TRIGger:A:BUS:B<x>:I2C:ADDRESS:TYPE?`

**Arguments**     `Generalcall` specifies a general call address.  
                  `STARTbyte` specifies a start byte address.  
                  `HSmode` specifies a high-speed mode address.  
                  `EEPROM` specifies an EEPROM address.  
                  `USER` specifies a user address.

## **TRIGger:A:BUS:B<x>:I2C:ADDRESS:VALUE**

This command sets or queries the binary address string used for the I2C trigger if the trigger condition is ADDR or ADDRANDDATA.

**Conditions**     This command requires a DPO4EMBD application module.

**Group**           Trigger

**Syntax**           `TRIGger:A:BUS:B<x>:I2C:ADDRESS:VALUE <NR3>`  
                      `TRIGger:A:BUS:B<x>:I2C:ADDRESS:VALUE?`

**Arguments**       `<NR3>` is up to 7 or 10-bits depending on the address mode that specifies the address.

**Examples**        `TRIGGER:A:BUS:B1:I2C:ADDRESS:VALUE 1011` sets the I2C address value to 1011.

## **TRIGger:A:BUS:B<x>:I2C:CONDition**

This command sets or returns the trigger condition for I2C trigger.

**Conditions**     This command requires a DPO4EMBD application module.

**Group**           Trigger

**Syntax**     `TRIGger:A:BUS:B<x>:I2C:CONDition`  
                   `{START|STOP|REPEATstart|ACKMISS|ADDRESS|DATA|ADDRANDDATA}`  
                   `TRIGger:A:BUS:B<x>:I2C:CONDition?`

**Arguments**     `START` specifies a search based on start condition.  
                       `STOP` specifies a search based on stop condition.  
                       `REPEATstart` specifies a search based on repeat of start condition.  
                       `ACKMISS` specifies a search based on missing acknowledgement condition.  
                       `ADDRESS` specifies a search based on address.  
                       `DATA` specifies a search based on data.  
                       `ADDRANDDATA` specifies a search based on address and data.

**Examples**     `TRIGGER:A:BUS:B1:I2C:CONDITION` `START` specifies start as the I2C trigger condition.

## TRIGger:A:BUS:B<x>:I2C:DATA:DIRection

This command sets or queries the I2C trigger condition valid on a READ, WRITE, or either. The read or write is indicated by the R/W bit in the I2C protocol.

**Conditions**     This command requires a DPO4EMBD application module.

**Group**           Trigger

**Syntax**        `TRIGger:A:BUS:B<x>:I2C:DATA:DIRection` `{READ|WRITE|NOCARE}`  
                       `TRIGger:A:BUS:B<x>:I2C:DATA:DIRection?`

**Arguments**     `READ` specifies read as the data direction.  
                       `WRITE` specifies write as the data direction.  
                       `NOCARE` specifies either as the data direction.

**Examples**     `TRIGGER:A:BUS:B1:I2C:DATA:DIRECTION` `WRITE` specifies write as the I2C data direction.



## TRIGger:A:BUS:B<x>:I2C:DATa:SIZE

This command sets or returns the length of the data string in bytes to be used for I2C trigger if the trigger condition is DATA or ADDRANDDATA.

**Conditions** This command requires a DPO4EMBD application module.

**Group** Trigger

**Syntax** TRIGger:A:BUS:B<x>:I2C:DATa:SIZE <NR3>  
TRIGger:A:BUS:B<x>:I2C:DATa:SIZE?

**Arguments** <NR3> is the length of the data string in bytes.

## TRIGger:A:BUS:B<x>:I2C:DATa:VALue

This command sets or queries the binary data string used for I2C triggering if the trigger condition is DATA or ADDRANDDATA.

**Conditions** This command requires a DPO4EMBD application module.

**Group** Trigger

**Syntax** TRIGger:A:BUS:B<x>:I2C:DATa:VALue <bin>  
TRIGger:A:BUS:B<x>:I2C:DATa:VALue?

**Arguments** <bin> is the binary data string with the number of bits specified by the [TRIGger:A:BUS:B<x>:I2C:DATa:SIZE](#) command.

## TRIGger:A:BUS:B<x>:SPI:CONDition

This command sets or queries the trigger condition for SPI triggering.

**Conditions** This command requires a DPO4EMBD application module.

**Group** Trigger

**Syntax**     `TRIGger:A:BUS:B<x>:SPI:CONDition {SS|MISO|MOSI|MISOMOSI}`  
`TRIGger:A:BUS:B<x>:SPI:CONDition?`

**Arguments**     SS specifies the Slave Selection condition

                  MISO specifies the Master-In Slave-Out condition.

                  MOSI specifies the Master-Out Slave-In condition.

                  MISOMOSI specifies the Master-In Slave-Out and Master-Out Slave-In conditions.

## **TRIGger:A:BUS:B<x>:SPI:DATA{IN|MISO}:VALue**

This command sets or returns the binary data string to be used for SPI trigger if trigger condition is MISO, MISOMOSI.

**Conditions**     This command requires a DPO4EMBD application module.

**Group**           Trigger

**Syntax**        `TRIGger:A:BUS:B<x>:SPI:DATA{:IN|MISO}:VALue <bin>`  
`TRIGger:A:BUS:B<x>:SPI:DATA{:IN|MISO}:VALue?`

**Arguments**     <bin> is the binary data string with the number of bits specified by the  
[TRIGger:A:BUS:B<x>:SPI:DATA:SIZE](#) command.

## **TRIGger:A:BUS:B<x>:SPI:DATA{OUT|MOSI}:VALue**

This command sets or queries the binary data string used for the SPI trigger if the trigger condition is set to MISO or MISOMOSI.

**Conditions**     This command requires a DPO4EMBD application module.

**Group**           Trigger

**Syntax**        `TRIGger:A:BUS:B<x>:SPI:DATA{:OUT|MOSI}:VALue <bin>`  
`TRIGger:A:BUS:B<x>:SPI:DATA{:OUT|MOSI}:VALue?`

**Arguments**     <bin> is the binary data string with the number of bits specified by the  
[TRIGger:A:BUS:B<x>:SPI:DATA:SIZE](#) command.

## TRIGger:A:BUS:B<x>:SPI:DATa:SIZE

This command sets or returns the length of the data string in bytes to be used for SPI trigger, if trigger condition is MISO, MOSI or MISOMOSI.

**Conditions** This command requires a DPO4EMBD application module.

**Group** Trigger

**Syntax** TRIGger:A:BUS:B<x>:SPI:DATa:SIZE <NR3>  
TRIGger:A:BUS:B<x>:SPI:DATa:SIZE?

**Arguments** <NR3> is the length of the data string in bytes.

## TRIGger:A:BUS:SOUrce

This command sets or queries the bus for a serial search.

**Conditions** This command requires a DPO4AUTO or DPO4EMBD application module.

**Group** Trigger

**Syntax** TRIGger:A:BUS:SOUrce {B1|B2}  
TRIGger:A:BUS:SOUrce?

**Arguments** B1 specifies the bus 1 source.  
B2 specifies the bus 2 source.

## TRIGger:A:EDGE? (Query Only)

This query-only command returns the trigger source, coupling, and slope for the A edge trigger. This command is equivalent to selecting Edge Setup from the Trig menu and viewing the current setups.

**Group** Trigger

**Syntax** TRIGger:A:EDGE?

**Related Commands**     [TRIGger:A:PULse?](#), [TRIGger:A:LOGic?](#)

**Examples**     `TRIGGER:A:EDGE?` might return `:TRIGGER:A:EDGE:SOURCE CH1;COUPLING DC; SLOPE RISE` indicating the trigger source, coupling, and slope for the A edge trigger.

## TRIGger:A:EDGE:COUPling

This command sets or returns the type of coupling for the A edge trigger. This command is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu, selecting Edge Trigger, and choosing from the Coupling drop-down list.

**Group**     Trigger

**Syntax**     `TRIGger:A:EDGE:COUPling {AC|DC|HFRej|LFRej|NOISerej}`  
`TRIGger:A:EDGE:COUPling?`

**Related Commands**     [TRIGger:A:EDGE:SOUrce](#), [TRIGger:A:EDGE:SLOpe](#)

**Arguments**     AC selects AC trigger coupling, which passes the input signals above 60 Hz to the trigger circuitry.

DC selects DC trigger coupling, which passes all input signals to the trigger circuitry.

HFRej coupling attenuates signals above 50 kHz before passing the signals to the trigger circuitry.

LFRej coupling attenuates signals below 80 kHz before passing the signals to the trigger circuitry.

NOISerej coupling provides stable triggering by increasing the trigger hysteresis. Increased hysteresis reduces the trigger sensitivity to noise but may require greater trigger signal amplitude.

**Examples**     `TRIGGER:A:EDGE:COUPLING DC` sets the A edge trigger coupling to DC.

`TRIGGER:A:EDGE:COUPLING?` might return `:TRIGGER:A:EDGE:COUPLING DC` indicating that the A edge trigger coupling is set to DC.

## TRIGger:A:EDGE:SLOpe

This command sets or returns the slope for the A edge trigger. This command is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu and then choosing the desired Slope.

**Group** Trigger

**Syntax** TRIGger:A:EDGE:SLOpe {RISe|FALL}  
TRIGger:A:EDGE:SLOpe?

**Related Commands** [TRIGger:A:EDGE:SOUrce](#), [TRIGger:A:EDGE:COUPling](#)

**Arguments** RISe specifies to trigger on the rising or positive edge of a signal.  
FALL specifies to trigger on the falling or negative edge of a signal.

**Examples** TRIGGER:A:EDGE:SLOPE RISE sets the A edge trigger slope to positive, which triggers on the rising edge of the signal.

TRIGGER:A:EDGE:SLOPE? might return :TRIGGER:A:EDGE:SLOPE FALL indicating that the A edge trigger slope is negative.

## TRIGger:A:EDGE:SOUrce

This command sets or returns the source for the A edge trigger. This command is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu and then choosing from the Source drop-down list.

**Group** Trigger

**Syntax** TRIGger:A:EDGE:SOUrce {CH<x>|EXT|LINE}  
TRIGger:A:EDGE:SOUrce?

**Related Commands** [TRIGger:A:EDGE:SLOpe](#), [TRIGger:A:EDGE:COUPling](#)

**Arguments** CH<x> specifies one input channel as the A edge trigger source.  
EXT specifies an external trigger using the Auxiliary Trigger Input connector located on the rear panel of the instrument.  
LINE specifies AC line voltage.

**Examples**     `TRIGGER:A:EDGE:SOURCE CH1` sets channel 1 as the A edge trigger source.  
`TRIGGER:A:EDGE:SOURCE?` might return `:TRIGGER:A:EDGE:SOURCE CH1` indicating that channel 1 is the A edge trigger source.

## TRIGger:A:HOLDoff? (Query Only)

This query-only command returns the A trigger holdoff parameters. These parameters specify the time period during which the trigger circuitry is not looking to generate a trigger event. This command is equivalent to selecting Holdoff from the Trig menu and then viewing the current settings.

**Group**     Trigger

**Syntax**     `TRIGger:A:HOLDoff?`

**Related Commands**     [TRIGger:A:HOLDoff:TIME](#)

**Examples**     `TRIGGER:A:HOLDOFF?` might return `:TRIGGER:A:HOLDOFF:TIME 900.0000E-09;BY DEFAULT` indicating that the A edge trigger holdoff time (by default) is 900 ns.

## TRIGger:A:HOLDoff:TIME

This command sets or returns the A trigger holdoff time. This command is equivalent to selecting Holdoff from the Trig menu and then choosing the desired Trig Holdoff.

**Group**     Trigger

**Syntax**     `TRIGger:A:HOLDoff:TIME <NR3>`  
`TRIGger:A:HOLDoff:TIME?`

**Arguments**     `<NR3>` specifies the holdoff time in seconds. The range is from 250 ns through 12.0 s.

**Examples**     `TRIGGER:A:HOLDOFF:TIME?` might return `:TRIGGER:A:HOLDOFFTIME 1.2000E-06` indicating that the A trigger holdoff time is set to 1.2  $\mu$ s.  
`TRIGGER:A:HOLDOFF:TIME 10` sets the A trigger holdoff time to 10 s.

## TRIGger:A:LEVel

This command sets or returns the level for the A trigger. This command is equivalent to selecting Level from the Trig menu and then viewing or setting the trigger Level.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:A:LEVel {ECL TTL <NR3>} TRIGger:A:LEVel?
<b>Arguments</b>	ECL specifies the ECL high level. TTL specifies the TTL high level. <NR3> specifies the trigger level in user units (usually volts).
<b>Examples</b>	TRIGGER:A:LEVEL? might return :TRIGGER:A:LEVel 1.3000E+00 indicating that the A edge trigger is set to 1.3 V. TRIGGER:A:LEVEL TTL sets the A edge trigger to TTL high level, which is 1.4 V.

## TRIGger:A:LEVel:CH<x>

This command sets or returns the level for the specified channel. Each Channel can have an independent Level.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:A:LEVel:CH<x> {<NR3> ECL TTL} TRIGger:A:LEVel:CH<x>?
<b>Arguments</b>	ECL specifies the ECL high level. TTL specifies the TTL high level. <NR3> specifies the trigger level in user units (usually volts).
<b>Examples</b>	TRIGGER:A:LEVEL:CH2? might return :TRIGGER:A:LEVEL:CH2 1.3000E+00 indicating that the A edge trigger is set to 1.3 V for channel 2. TRIGGER:A:LEVEL:CH3 TTL sets the A edge trigger to TTL high level for channel 3.

## TRIGger:A:LOGIc? (Query Only)

This query-only command returns all of the A logic trigger parameters. This command is provided for backward compatibility with other Tektronix instruments.

**Group** Trigger

**Syntax** TRIGger:A:LOGIc?

**Related Commands** [TRIGger:A:LOGIc:CLAss](#)

**Examples** TRIGGER:A:LOGIC? might return :TRIGGER:A:LOGIC:CLASS PATTERN; FUNCTION AND;WHEN TRUE; THRESHOLD:CH1 1.4000;CH2 1.4000;CH3 1.4000; CH4 1.4000;;TRIGGER:A:LOGIC:INPUT:CH1 HIGH; CH2 X;CH3 X; :TRIGGER:A:LOGIC:PATTERN:INPUT:CH4 X; :TRIGGER:A:LOGIC:PATTERN:WHEN TRUE; WHEN:LESSLIMIT 5.0000E-9; MORELIMIT 5.0000E-9; :TRIGGER:A:LOGIC:SETHOLD:CLOCK:EDGE RISE; THRESHOLD 1.4000;SOURCE CH2; :TRIGGER:A:LOGIC:SETHOLD:DATA:THRESHOLD 1.4000;SOURCE CH1; :TRIGGER:A:LOGIC:SETHOLD:HOLDTIME2.0000E-9; SETTIME 3.0000E-9; :TRIGGER:A:LOGIC:STATE:INPUT:CH4 RISE; :TRIGGER:A:LOGIC:STATE:WHEN TRUE

## TRIGger:A:LOGIc:CLAss

This command sets or returns the class of the Logic Trigger. Used in conjunction with the [TRIGger:A:TYPe](#) command, this command is equivalent to selecting Logic Pattern, Logic State or Setup/Hold Setup from the Trig menu. This command is provided for backward compatibility with other Tektronix instruments.

**Group** Trigger

**Syntax** TRIGger:A:LOGIc:CLAss {PATtern|STATE|SETHo1d}  
TRIGger:A:LOGIc:CLAss?

**Related Commands** [TRIGger:A:TYPe](#), [TRIGger:A:PULse:CLAss](#)



**Arguments** **PATtern** sets the instrument to trigger when the specified logical combinations of channels 1, 2, 3, and 4 are met on four-channel instruments. On two-channel instruments, only channel 1 and channel 2 are available.

**STATE** sets the instrument to trigger when the specified conditions of channels 1, 2, and 3 are met after the channel 4 (clock) condition is met on four-channel instruments. On two-channel instruments, only channel 1 and channel 2 (clock) are available.

**SETHold** sets the instrument to trigger on setup and hold violations between a data source and a clock source. Use one channel input as the clock signal and a second channel input as the data input. The clocking and data levels are used to determine if a clock or data transition has occurred.

**Examples** **TRIGGER:A:LOGIC:CLASS?** might return **:TRIGGER:A:LOGIC:CLASS PATTERN**

**TRIGGER:A:LOGIC:CLASS PATTERN** sets the trigger A logic class to **PATtern**, which causes the instrument to trigger when the specified logical combinations of channels 1, 2, 3, and 4 are met.

## TRIGger:A:LOGic:FUNcTion

This command sets or returns the logical combination of the input channels for the A pattern and A state logic triggers. This command is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu, selecting Pattern or State for the Trigger Type, and setting or viewing the Pattern Type. This command is provided for backward compatibility with other Tektronix instruments.

**Group** Trigger

**Syntax** **TRIGger:A:LOGic:FUNcTion {AND|NAND|NOR|OR}**  
**TRIGger:A:LOGic:FUNcTion?**

**Related Commands** [TRIGger:A:LOGic:PATtern:WHEn](#), [TRIGger:A:LOGic:INPut:CH<x>](#),  
[TRIGger:A:LOGic:THReshold:CH<x>](#)

**Arguments** **AND** specifies to trigger if all conditions are true.

**NAND** specifies to trigger if any of the conditions are false.

**NOR** specifies to trigger if all conditions are false.

**OR** specifies to trigger if any of the conditions are true.

<b>Examples</b>	<p><code>TRIGGER:A:LOGIC:FUNCTION?</code> might return <code>:TRIGGER:A:LOGIC:FUNCTION NAND</code></p> <p>which indicates that the instrument will trigger if the AND logic conditions are false.</p> <p><code>TRIGGER:A:LOGIC:FUNCTION AND</code> sets the logical combination of channels to be true when all conditions are true.</p>
-----------------	--

## TRIGger:A:LOGic:INPut? (Query Only)

On four-channel instruments, this query-only command returns the A logic trigger input expected for channel 1, 2, and 3. Channel 4 is set or queried with the command [TRIGger:A:LOGic:PATtern:INPut:CH4](#).

On two-channel instruments, this query returns the A logic trigger input expected for channel 1. Channel 2 is set or queried with the command [TRIGger:A:LOGic:PATtern:INPut:CH2](#).

This command is equivalent to selecting A Event (Main) Trigger Setup and viewing or setting the Input Threshold for channel 1 through 3 for four-channel instruments or channel 1 for two-channel instruments.

<b>Group</b>	Trigger
<b>Syntax</b>	<code>TRIGger:A:LOGic:INPut?</code>
<b>Related Commands</b>	<a href="#">TRIGger:A:LOGic:PATtern:INPut:CH4</a> , <a href="#">TRIGger:A:LOGic:PATtern:INPut:CH2</a>
<b>Examples</b>	<p><code>TRIGGER:A:LOGIC:INPUT?</code> might return <code>:TRIGGER:A:LOGIC:INPUT:CH1 HIGH;CH2 X;CH3 X</code> indicating that a logic high is expected on channel 1 while channel 2 and channel three are “don’t care.”</p>

## TRIGger:A:LOGic:INPut:CH<x>

This command sets or returns the A logical input for the logic trigger channel specified by x. The value of x ranges from 1 through 3 for four-channel instruments. For two-channel instruments x can only be 1. Note that CH4 on four-channel instruments or CH2 on two-channel instruments cannot be set or queried with this command. For details about setting this channel, see [TRIGger:A:LOGic:PATtern:INPut:CH4](#) or [TRIGger:A:LOGic:PATtern:INPut:CH2](#). This command is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu and then choosing the desired

logical input from the Ch<x> drop-down list, which is located in the Input Threshold group box.

<b>Group</b>	Trigger
<b>Syntax</b>	<pre>TRIGger:A:LOGic:INPut:CH&lt;x&gt; {HIGH LOW X} TRIGger:A:LOGic:INPut:CH&lt;x&gt;?</pre>
<b>Related Commands</b>	<a href="#">TRIGger:A:LOGic:PATtern:INPut:CH4</a> , <a href="#">TRIGger:A:LOGic:PATtern:INPut:CH2</a>
<b>Arguments</b>	<p>HIGH specifies the logic high.</p> <p>LOW specifies the logic low.</p> <p>X specifies a "don't care" state.</p>
<b>Examples</b>	<p>TRIGGER:A:LOGIC:INPUT:CH&lt;x&gt;? might return :TRIGGER:LOGIC:INPUT:CH1 X indicating that the setting for the A logic trigger input to channel 1 does not matter.</p> <p>TRIGGER:A:LOGIC:INPUT:CH&lt;x&gt; HIGH sets the A logic trigger input to logic HIGH for channel 2.</p>

## TRIGger:A:LOGic:INPut:CLOCK:EDGE

Sets the polarity of the clock channel.

<b>Group</b>	Trigger
<b>Syntax</b>	<pre>TRIGger:A:LOGic:INPut:CLOCK:EDGE {FALL RISe} TRIGger:A:LOGic:INPut:CLOCK:EDGE?</pre>
<b>Arguments</b>	<p>RISe specifies to trigger on the rising or positive edge of a signal.</p> <p>FALL specifies to trigger on the falling or negative edge of a signal.</p>

## TRIGger:A:LOGic:INPut:CLOCK:SOUrce

Sets or returns channel to use as the clock source. The Clock can be selected as 'NONE'. A selection of none implies pattern trigger. Any other selection implies state trigger.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:A:LOGic:INPut:CLOCK:SOURce {CH1 CH2 CH3 CH4 NONE} TRIGger:A:LOGic:INPut:CLOCK:SOURce?
<b>Arguments</b>	CH<x> specifies the input channel, which ranges from 1 through 4 for four-channel instruments or 1 through 2 for two channel instruments.  NONE specifies a Pattern trigger.

## TRIGger:A:LOGic:PATtern? (Query Only)

This query-only command returns the conditions used for generating an A logic pattern trigger, with respect to the defined input pattern, and identifies the maximum and minimum time that the selected pattern may be true and still generate the trigger. This command is equivalent to selecting Logic Pattern from the Trig menu and then viewing the current setups.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:A:LOGic:PATtern?
<b>Related Commands</b>	<a href="#">TRIGger:A:LOGic:PATtern:INPut:CH4</a> , <a href="#">TRIGger:A:LOGic:PATtern:WHEn</a> , <a href="#">TRIGger:A:LOGic:PATtern:WHEn:LESSLimit</a> , <a href="#">TRIGger:A:LOGic:PATtern:WHEn:MORELimit</a>
<b>Examples</b>	TRIGGER:A:LOGIC:PATTERN? might return :TRIGGER:A:LOGIC:PATTERN:INPUT:CH4 X; :TRIGGER:A:LOGIC:PATTERN:WHEN TRUE; WHEN:LESSLIMIT 5.0000E-9;MORELIMIT 5.0000E-9.

## TRIGger:A:LOGic:PATtern:INPut:CH2

This command sets or returns the A logic trigger input for channel 2 in two-channel instruments only. The command specifies the logic value used when the pattern trigger detects the threshold level. Sending this command is equivalent to selecting Logic Pattern from the Trig menu and then choosing the desired logical input from the Ch2 drop-down list, which is located in the Input Threshold group box.

To set or query the A logic trigger input for a four-channel instrument, see [TRIGger:A:LOGic:PATtern:INPut:CH4](#).

**Conditions** This command applies to two-channel instruments.

**Group** Trigger

**Syntax** TRIGGER:A:LOGIC:PATTERN:INPUT:CH2 {HIGH|LOW|X}  
TRIGGER:A:LOGIC:PATTERN:INPUT:CH2?

**Related Commands** [TRIGGER:A:LOGIC:FUNCTION](#), [TRIGGER:A:LOGIC:INPUT:CH<x>](#),  
[TRIGGER:A:LOGIC:THRESHOLD:CH<x>](#)

**Arguments** HIGH specifies the logic high.  
LOW specifies the logic low.  
X specifies a "don't care" state.

**Examples** TRIGGER:A:LOGIC:PATTERN:INPUT:CH2 HIGH sets the A logic trigger input to logic high for channel 2 when the logic class is set to PATTERN. When the threshold level is detected, HIGH places a 1 on the channel 2 input to the selected function.  
  
TRIGGER:A:LOGIC:PATTERN:INPUT:CH2? might return  
:TRIGGER:A:LOGIC:PATTERN:INPUT:CH2 HIGH indicating that the logic input for channel 2 is logic high.

## TRIGGER:A:LOGIC:PATTERN:INPUT:CH4

This command sets or returns the A logic trigger input for channel 4 in four-channel instruments only. The command specifies the logic value used when the pattern trigger detects the threshold level. Sending this command is equivalent to selecting Logic Pattern from the Trig menu and then choosing the desired logical input from the Ch4 drop-down list, which is located in the Input Threshold group box.

To set or query the A logic trigger input for a two-channel instrument, see [TRIGGER:A:LOGIC:PATTERN:INPUT:CH2](#).

**Conditions** This command applies to four-channel instruments.

**Group** Trigger

<b>Syntax</b>	<pre>TRIGger:A:LOGic:PATtern:INPut:CH4 {HIGH LOW X} TRIGger:A:LOGic:PATtern:INPut:CH4?</pre>
<b>Related Commands</b>	<a href="#">TRIGger:A:LOGic:FUNcTion</a> , <a href="#">TRIGger:A:LOGic:INPut:CH&lt;x&gt;</a> , <a href="#">TRIGger:A:LOGic:THReshold:CH&lt;x&gt;</a>
<b>Arguments</b>	<p>HIGH specifies the logic high.</p> <p>LOW specifies the logic low.</p> <p>X specifies a "don't care" state.</p>
<b>Examples</b>	<p>TRIGGER:A:LOGIC:PATTERN:INPUT:CH4 HIGH sets the A logic trigger input to logic high for channel 4 when the logic class is set to PATtern. When the threshold level is detected, HIGH places a 1 on the channel 4 input to the selected function.</p> <p>TRIGGER:A:LOGIC:PATTERN:INPUT:CH4? might return  :TRIGGER:A:LOGIC:PATTERN:INPUT:CH4 HIGH indicating that the logic input for channel 4 is logic high.</p>

## TRIGger:A:LOGic:PATtern:WHEN

This command sets or returns the condition for generating an A logic pattern trigger with respect to the defined input pattern. This command is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu, selecting Pattern for Trigger Type, and choosing a trigger condition from the Pattern drop-down list, which is located in the Trigger When group box.

<b>Group</b>	Trigger
<b>Syntax</b>	<pre>TRIGger:A:LOGic:PATtern:WHEN {TRUE FALSE LESSThan MOREThan EQUal UNEQUal} TRIGger:A:LOGic:PATtern:WHEN?</pre>
<b>Related Commands</b>	<a href="#">TRIGger:A:LOGic:FUNcTion</a> , <a href="#">TRIGger:A:LOGic:INPut?</a> , <a href="#">TRIGger:A:LOGic:THReshold?</a> , <a href="#">TRIGger:A:LOGic:PATtern:WHEn:LESSLimit</a> , <a href="#">TRIGger:A:LOGic:PATtern:WHEn:MORELimit</a>
<b>Arguments</b>	<p>TRUE argument sets the instrument to trigger when the pattern becomes true.</p> <p>FALSE argument sets the instrument to trigger when the pattern becomes false.</p>

LESSThan argument sets the instrument to trigger if the specific pattern is true less than the time set by the [TRIGger:A:LOGic:PATtern:WHEn:LESSLimit](#) command.

MOREThan argument sets the instrument to trigger if the specific pattern is true longer than the specified time set by the [TRIGger:A:LOGic:PATtern:WHEn:MORELimit](#) command.

**Examples** `TRIGGER:A:LOGIC:PATTERN:WHEN FALSE` specifies to trigger the A logic pattern when the pattern becomes false.

`TRIGGER:A:LOGIC:PATTERN:WHEN?` might return  
`:TRIGGER:A:LOGIC:PATTERN:WHEN TRUE` indicating that the A logic pattern will trigger when the pattern becomes true.

## TRIGger:A:LOGic:PATtern:WHEn:LESSLimit

This command sets or returns the maximum time that the selected pattern may be true and still generate an A logic pattern trigger. This command is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu, selecting Pattern as the Trigger Type, selecting Less Than for the Pattern in the Trigger When settings, and entering a maximum value for Time.

**Group** Trigger

**Syntax** `TRIGger:A:LOGic:PATtern:WHEn:LESSLimit <NR3>`  
`TRIGger:A:LOGic:PATtern:WHEn:LESSLimit?`

**Related Commands** [TRIGger:A:LOGic:PATtern:WHEn:MORELimit](#)

**Arguments** `<NR3>` specifies the amount of time to hold the pattern true.

**Examples** `TRIGGER:A:LOGIC:PATTERN:WHEN:LESSLIMIT 10.0E+00` sets the maximum time that the selected pattern may hold true (and generate an A logic pattern trigger) to 10 s.

`TRIGGER:A:LOGIC:PATTERN:WHEN:LESSLIMIT?` might return  
`:TRIGGER:A:LOGIC:PATTERN:WHEN:LESSLIMIT 5.0000E-09` indicating that the selected pattern may hold true for up to 5 ns and still generate an A logic pattern trigger.

## TRIGger:A:LOGIc:PATtern:WHEn:MORELimit

This command sets or returns the minimum time that the selected pattern may be true and still generate an A logic pattern trigger. This command is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu, selecting Pattern as the Trigger Type, selecting More Than for the Pattern in the Trigger When settings, and entering a minimum value for Time.

**Group** Trigger

**Syntax** TRIGger:A:LOGIC:PATtern:WHEn:MORELimit <NR3>  
TRIGger:A:LOGIC:PATtern:WHEn:MORELimit?

**Related Commands** [TRIGger:A:LOGIc:PATtern:WHEn:LESSLimit](#)

**Arguments** <NR3> specifies the amount of time to hold the pattern true.

**Examples** TRIGGER:A:LOGIC:PATTERN:WHEN:MORELIMIT 10.0E+00 sets the minimum time that the selected pattern may hold true (and generate an A logic pattern trigger) to 10 s.

TRIGGER:A:LOGIC:PATTERN:WHEN:MORELIMIT? might return  
:TRIGGER:A:LOGIC:PATTERN:WHEN:MORELIMIT 5.0000E-09 indicating that the selected pattern must hold true for at least 5 ns to generate an A logic pattern trigger.

## TRIGger:A:LOGIc:SETHold:CLOCK:EDGE

This command sets or returns the clock edge polarity for setup and hold triggering. This command is provided for compatibility with other Tektronix instruments. See the [TRIGger:A:SETHold:CLOCK:EDGE](#) command.

**Group** Trigger

**Syntax** TRIGger:A:LOGIC:SETHold:CLOCK:EDGE?

## TRIGger:A:LOGIc:SETHold:CLOCK:SOUrce

This command sets or returns the clock source for the A logic trigger setup and hold input. This command is provided for compatibility with other Tektronix instruments. See the [TRIGger:A:SETHold:CLOCK:SOUrce](#) command.



**Group** Trigger

**Syntax** TRIGger:A:LOGic:SETHold:CLOCK:SOUrce

## TRIGger:A:LOGic:SETHold:CLOCK:THReshold

This command sets or returns the clock voltage threshold for the setup and hold trigger. This command is provided for compatibility with other Tektronix instruments. See the [TRIGger:A:SETHold:CLOCK:THReshold](#) command.

**Group** Trigger

**Syntax** TRIGger:A:LOGic:SETHold:CLOCK:THReshold {ECL|TTL|<NR3>}  
TRIGger:A:LOGic:SETHold:CLOCK:THReshold?

## TRIGger:A:LOGic:SETHold:DATA:SOUrce

This command sets or returns the data source for the setup and hold trigger. This command is provided for compatibility with other Tektronix instruments. See the [TRIGger:A:SETHold:DATA:SOUrce](#) command.

**Group** Trigger

**Syntax** TRIGger:A:LOGic:SETHold:DATA:SOUrce

## TRIGger:A:LOGic:SETHold:DATA:THReshold

This command sets or returns the data voltage threshold for setup and hold trigger. This command is provided for compatibility with other Tektronix instruments. See the [TRIGger:A:SETHold:DATA:THReshold](#) command.

**Group** Trigger

**Syntax** TRIGger:A:LOGic:SETHold:DATA:THReshold {ECL|TTL|<NR3>}  
TRIGger:A:LOGic:SETHold:DATA:THReshold?

**Related Commands** [TRIGger:A:LOGic:SETHold:DATA:SOUrce](#)

<b>Arguments</b>	<p>ECL specifies the preset ECL high level of -1.3 V.</p> <p>TTL specifies the preset TTL high level of 1.4 V.</p> <p>&lt;NR3&gt; is the setup and hold data level, in V.</p>
<b>Examples</b>	<p>TRIGGER:A:LOGIC:SETHOLD:DATA:THRESHOLD TTL specifies the preset ECL high level of 1.4 V as the current data voltage level for the setup and hold trigger.</p> <p>TRIGGER:A:LOGIC:SETHOLD:DATA:THRESHOLD? might return :TRIGGER:A:LOGIC:SETHOLD:DATA:THRESHOLD 1.2000E+00 indicating that 1.2 V is the current data voltage level for the setup and hold trigger.</p>

## TRIGger:A:LOGic:SETHold:HOLDTime

This command sets or returns the hold time for setup and hold violation triggering. This command is provided for compatibility with other Tektronix instruments. See the [TRIGger:A:SETHold:HOLDTime](#) command.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:A:LOGic:SETHold:HOLDTime

## TRIGger:A:LOGic:SETHold:SETTime

This command sets or returns the setup time for setup and hold violation triggering. This command is provided for compatibility with other Tektronix instruments. See the [TRIGger:A:SETHold:SETTime](#) command.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:A:LOGic:SETHold:SETTime

## TRIGger:A:LOGic:THReshold? (Query Only)

This query-only command returns the threshold voltage for all channels in an A logic trigger. This command is provided for compatibility with other Tektronix instruments. See the [TRIGger:A:LEVel](#) command

<b>Group</b>	Trigger
--------------	---------

**Syntax**    `TRIGger:A:LOGic:THReshold?`

## **TRIGger:A:LOGic:THReshold:CH<x>**

This command sets or returns the A logic trigger threshold voltage for the channel, specified by x, which ranges from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. See the [TRIGger:A:LEVel:CH<x>](#) command

**Group**    Trigger

**Syntax**    `TRIGger:A:LOGic:THReshold:CH<x>`

## **TRIGger:A:LOWerthreshold:CH<x>**

This command sets or returns the lower threshold for the channel selected. Each channel can have an independent level. Used in runt and slew rate as the lower threshold. Used for all other trigger types as the single level/threshold.

**Group**    Trigger

**Syntax**    `TRIGger:A:LOWerthreshold:CH<x> {<NR3>|ECL|TTL}`  
`TRIGger:A:LOWerthreshold:CH<x>?`

**Arguments**    ECL specifies a preset ECL high level of -1.3 V.  
TTL specifies a preset TTL high level of 1.4 V.  
<NR3> is the clock level, in volts.

**Examples**    `TRIGGER:A:LOWERTHRESHOLD:CH2 50E-3` sets the lower limit of the pulse runt trigger to 50 mV for channel 2.  
`TRIGGER:A:LOWERTHRESHOLD:CH2?` might return `:TRIGGER:A:LOWERTHRESHOLD:CH2 1.2000E-01` indicating that the lower limit of the pulse runt trigger is set to 120 mV.

## TRIGger:A:MODE

This command sets or returns the A trigger mode. This command is equivalent to selecting Mode from the Trig menu and then choosing the desired Trigger Mode.

**Group** Trigger

**Syntax** TRIGger:A:MODE {AUTO|NORMa1}  
TRIGger:A:MODE?

**Related Commands** [TRIGger:A:LEVel](#)

**Arguments** AUTO generates a trigger if one is not detected within a specified time period.  
NORMa1 waits for a valid trigger event.

**Examples** TRIGGER:A:MODE NORMAL specifies that a valid trigger event must occur before a trigger is generated.  
  
TRIGGER:A:MODE? might return :TRIGGER:A:MODE NORMAL indicating that a valid trigger event must occur before a trigger is generated.

## TRIGger:A:PULse? (Query Only)

This query-only command returns the A pulse trigger parameters.

**Group** Trigger

**Syntax** TRIGger:A:PULse?

**Related Commands** [TRIGger:A:EDGE?](#), [TRIGger:A:LOGIc?](#)

**Examples** TRIGGER:A:PULSE? might return :TRIGGER:A:PULSE:CLASS  
GLITCH;SOURCE CH1; GLITCH:WIDTH 2.0000E-9; TRIGIF  
ACCEPT;POLARITY POSITIVE; :TRIGGER:A:PULSE:RUNT:POLARITY  
POSITIVE; THRESHOLD:HIGH 1.2000;LOW 800.0000E-3;  
:TRIGGER:A:PULSE:RUNT:WHEN OCCURS; WIDTH 2.0000E-9;  
:TRIGGER:A:PULSE:TRANSITION:DELTATIME 2.0000E-9;POLARITY  
POSITIVE; THRESHOLD:HIGH 1.2000;LOW 800.0000E-3;  
:TRIGGER:A:PULSE:TRANSITION:WHEN SLOWERTHAN;  
:TRIGGER:A:PULSE:WIDTH:LOWLIMIT 2.0000E-9;

```
HIGHLIMIT 2.0000E-9;WHEN WITHIN; POLARITY
POSITIVE;;:TRIGGER:A:PULSE:TIMEOUT:POLARITY STAYSHIGH;TIME
2.0000E-9
```

## TRIGger:A:PULse:CLAss

This command sets or returns the type of pulse on which to trigger. This command is equivalent to selecting the setup menu for the pulse type that you want from the Trig menu: Width Setup, Runt Setup, or Transition Setup.

**Group** Trigger

**Syntax** TRIGger:A:PULse:CLAss {RUnT|WIDTh|TRANSition}  
TRIGger:A:PULse:CLAss?

**Related Commands** [TRIGger:A:RUnT?](#), [TRIGger:A:PULse:WIDTh?](#), [TRIGger:A{:TRANSition|RISEFall}?](#), [TRIGger:A:TYPe](#)

**Arguments** **RUnT** triggers when a pulse crosses the first preset voltage threshold but does not cross the second preset threshold before recrossing the first. The thresholds are set with the [TRIGger:A:PULse:RUnT:THReshold:LOW](#) and [TRIGger:A:PULse:RUnT:THReshold:HIGh](#) commands.

**WIDTh** triggers when a pulse is found that has the specified polarity and is either inside or outside the limits as specified by [TRIGger:A:PULse:WIDTh:LOWLimit](#) and [TRIGger:A:PULse:WIDTh:HIGHLimit](#). The polarity is selected using the [TRIGger:A:PULse:WIDTh:POLarity](#) command.

**TRANSition** triggers when a pulse crosses both thresholds in the same direction as the specified polarity and the transition time between the two threshold crossings is greater or less than the specified time delta.

**Examples** `TRIGGER:A:PULSE:CLASS WIDTH` specifies a width pulse for the A trigger.

`TRIGGER:A:PULSE:CLASS?` might return `:TRIGGER:A:PULSE:CLASS WIDTH` indicating that a pulse was found that is of the specified polarity and width.

## TRIGger:A:PULse:RUnT:HIGHLimit

This command sets or returns the upper limit for the runt trigger. This command is provided for compatibility with other Tektronix instruments. See the [TRIGger:A:UPPerthreshold:CH<x>](#) command.

**Group** Trigger

**Syntax** TRIGger:A:PULse:RUNT:HIGHLimit

## TRIGger:A:PULse:RUNT:LOWLimit

This command sets or returns the lower limit for the runt trigger. This command is provided for compatibility with other Tektronix instruments. See the [TRIGger:A:LOWerthreshold:CH<x>](#) command.

**Group** Trigger

**Syntax** TRIGger:A:PULse:RUNT:LOWLimit

## TRIGger:A:PULse:RUNT:POLarity

This command sets or returns the polarity for the runt trigger. This command is provided for compatibility with other Tektronix instruments. See the [TRIGger:A:RUNT:POLarity](#) command.

**Group** Trigger

**Syntax** TRIGger:A:PULse:RUNT:POLarity

## TRIGger:A:PULse:RUNT:SOURce

This command sets or returns the source for the A runt trigger. This command is provided for compatibility with other Tektronix instruments. See the [TRIGger:A:RUNT:SOURce](#) command.

**Group** Trigger

**Syntax** TRIGger:A:PULse:RUNT:SOURce

## TRIGger:A:PULse:RUNT:THReshold? (Query Only)

This query-only command returns the upper and lower thresholds for the runt trigger. This command is provided for compatibility with other Tektronix instruments.

**Group** Trigger

**Syntax** TRIGger:A:PULse:RUNT:THReshold?

## TRIGger:A:PULse:RUNT:THReshold:HIGH

This command sets or returns the upper limit for the runt trigger. This command is provided for compatibility with other Tektronix instruments. See the [TRIGger:A:UPPerthreshold:CH<x>](#) command.

**Group** Trigger

**Syntax** TRIGger:A:PULse:RUNT:THReshold:HIGH

## TRIGger:A:PULse:RUNT:THReshold:LOW

This command sets or returns the lower limit for the A pulse runt trigger. This command is provided for compatibility with other Tektronix instruments. See the [TRIGger:A:LOWerthreshold:CH<x>](#) command.

**Group** Trigger

**Syntax** TRIGger:A:PULse:RUNT:THReshold:LOW

## TRIGger:A:PULse:RUNT:WHEn

This command sets or returns the type of pulse width the trigger checks for when it detects a runt. This command is provided for compatibility with other Tektronix instruments. See the [TRIGger:A:RUNT:WHEn](#) command.

**Group** Trigger

**Syntax**     `TRIGger:A:PULse:RUNT:WHEn`

## TRIGger:A:PULse:RUNT:WIDth

This command sets or returns the minimum width for a runt trigger. This command is provided for compatibility with other Tektronix instruments. See the [TRIGger:A:RUNT:WIDth](#) command.

**Group**     Trigger

**Syntax**     `TRIGger:A:PULse:RUNT:WIDth <NR3>`  
`TRIGger:A:PULse:RUNT:WIDth?`

**Related Commands**     [TRIGger:A:PULse:RUNT:WHEn](#)

**Arguments**     `<NR3>` specifies the minimum width, in seconds.

**Examples**     `TRIGGER:A:PULSE:RUNT:WIDTH 15E-6` sets the minimum width of the pulse runt trigger to 15  $\mu$ s.

`TRIGGER:A:PULSE:RUNT:WIDTH?` might return  
`:TRIGGER:A:PULSE:RUNT:WIDTH 2.0000E-09` indicating that  
the minimum width of a pulse runt trigger is 2 ns.

## TRIGger:A:PULse{:TRANsition|:SLEWRate}:DELtAtime

This command sets or returns the delta time used in calculating the transition value for the transition trigger. This command is provided for compatibility with other Tektronix instruments. See the [TRIGger:A:UPPerthreshold:CH<x>](#) command.

**Group**     Trigger

**Syntax**     `TRIGger:A:PULse{:TRANsition|:SLEWRate}:DELtAtime <NR3>`  
`TRIGger:A:PULse{:TRANsition|:SLEWRate}:DELtAtime?`

**Related Commands**     [TRIGger:A:PULse{:TRANsition|:SLEWRate}:POLarity](#), [TRIGger:A:PULse{:TRANsition|:SLEWRate}:THReshold?](#)



## TRIGger:A:PULse{:TRANSition|:SLEWRate}:HIGHLimit

This command sets or returns the upper (most positive) transition trigger threshold. This command is provided for compatibility with other Tektronix instruments. See the [TRIGger:A:UPPerthreshold:CH<x>](#) command.

**Group** Trigger

**Syntax** TRIGger:A:PULse{:TRANSition|:SLEWRate}:HIGHLimit

## TRIGger:A:PULse{:TRANSition|:SLEWRate}:LOWLimit

This command sets or returns the lower (most negative) transition trigger threshold. This command is provided for compatibility with other Tektronix instruments. See the [TRIGger:A:LOWerthreshold:CH<x>](#) command.

**Group** Trigger

**Syntax** TRIGger:A:PULse{:TRANSition|:SLEWRate}:LOWLimit

## TRIGger:A:PULse{:TRANSition|:SLEWRate}:POLarity

This command sets or returns the polarity for the transition trigger. This command is provided for compatibility with other Tektronix instruments. See the [TRIGger:A{:TRANSition|:RISEFall}:POLarity](#) command.

**Group** Trigger

**Syntax** TRIGger:A:PULse{:TRANSition|:SLEWRate}:POLarity  
{POSitive|NEGative|EITHer}  
TRIGger:A:PULse{:TRANSition|:SLEWRate}:POLarity?

## TRIGger:A:PULse{:TRANSition|:SLEWRate}:SOURce

This command sets or returns the source for transition trigger. This command is provided for compatibility with other Tektronix instruments. See the [TRIGger:A{:TRANSition|:RISEFall}:SOURce](#) command.

**Group** Trigger

**Syntax**    `TRIGger:A:PULse{:TRANSition|:SLEWRate}:SOURCE`

## **TRIGger:A:PULse{:TRANSition|:SLEWRate}:THReshold? (Query Only)**

This query-only command returns the upper and lower threshold limits for the transition time trigger. This command is provided for compatibility with other Tektronix instruments.

**Group**    Trigger

**Syntax**    `TRIGger:A:PULse{:TRANSition|:SLEWRate}:THReshold?`

**Examples**    `TRIGGER:A:PULSE{:TRANSITION|:SLEWRATE}:THRESHOLD?` might return `:TRIGGER:A:PULSE:TRANSITION:THRESHOLD:HIGH 1.2000;LOW 800.0000E-3` indicating the upper and lower threshold limits for the transition time trigger.

## **TRIGger:A:PULse{:TRANSition|:SLEWRate}:THReshold:HIGH**

This command sets or returns the upper (most positive) transition trigger threshold. This command is provided for compatibility with other Tektronix instruments. See the [TRIGger:A:UPPerthreshold:CH<x>](#) command.

**Group**    Trigger

**Syntax**    `TRIGger:A:PULse{:TRANSition|:SLEWRate}:THReshold:HIGH`

## **TRIGger:A:PULse{:TRANSition|:SLEWRate}:THReshold:LOW**

This command sets or returns the lower (most negative) transition trigger threshold. This command is provided for compatibility with other Tektronix instruments. See the [TRIGger:A:LOWerthreshold:CH<x>](#) command.

**Group**    Trigger

**Syntax**    `TRIGger:A:PULse{:TRANSition|:SLEWRate}:THReshold:LOW`

## TRIGger:A:PULse{:TRANSition|:SLEWRate}:WHEn

This command sets or returns whether to check for a transitioning signal that is faster or slower than the specified delta time. This is equivalent to selecting Transition Setup from the Trig menu and choosing the Trigger When Transition Time setting.

**Group** Trigger

**Syntax** TRIGger:A:PULse{:TRANSition|:SLEWRate}:WHEn  
{FASTER|SLOWER|EQUAL|UNEQUAL}  
TRIGger:A:PULse{:TRANSition|:SLEWRate}:WHEn?

## TRIGger:A:PULse:WIDth:HIGHLimit

This command sets or returns the upper limit for the width trigger. This command is provided for compatibility with other Tektronix instruments. See the [TRIGger:A:PULSEWidth:WIDth](#) command.

**Group** Trigger

**Syntax** TRIGger:A:PULse:WIDth:HIGHLimit

## TRIGger:A:PULse:WIDth:LEVel

This command sets or returns the threshold for the pulse width trigger. This command is provided for compatibility with other Tektronix instruments. See the [TRIGger:A:LEVel:CH<x>](#) command.

**Group** Trigger

**Syntax** TRIGger:A:PULse:WIDth:LEVel

## TRIGger:A:PULse:WIDth:LOWLimit

This command sets or returns the lower limit for the width trigger. This command is provided for compatibility with other Tektronix instruments. See the [TRIGger:A:PULSEWidth:WIDth](#) command.

**Group** Trigger

**Syntax** TRIGger:A:PULse:WIDTH:LOWLimit

## TRIGger:A:PULse:WIDTH:POLarity

This command sets or returns the polarity for the width trigger. This command is provided for compatibility with other Tektronix instruments. See the [TRIGger:A:PULSEWidth:POLarity](#) command.

**Group** Trigger

**Syntax** TRIGger:A:PULse:WIDTH:POLarity

## TRIGger:A:PULse:WIDTH:SOURce

This command sets or returns the source for the pulse width trigger. This command is provided for compatibility with other Tektronix instruments. See the [TRIGger:A:PULSEWidth:SOURce](#) command.

**Group** Trigger

**Syntax** TRIGger:A:PULse:WIDTH:SOURce {CH1|CH2|CH3|CH4|LINE|EXT}  
TRIGger:A:PULse:WIDTH:SOURce?

**Arguments** CH<x> specifies one input channel as the A edge trigger source.

EXT specifies an external trigger using the Auxiliary Trigger Input connector located on the rear panel of the instrument.

LINE specifies AC line voltage.

**Examples** TRIGGER:A:PULSE:WIDTH:SOURCE CH1 sets channel 1 as the pulse width source.

TRIGGER:A:PULSE:WIDTH:SOURCE? might return :TRIGGER:A:PULSE:WIDTH:SOURCE CH1 indicating that channel 1 is the pulse width source.

## TRIGger:A:PULse:WIDth:WHEn

This command sets or returns whether to trigger on a pulse width that falls outside (or within) the specified range of limits. This command is provided for compatibility with other Tektronix instruments. See the [TRIGger:A:PULSEWidth:WHEn](#) command.

**Group** Trigger

**Syntax** TRIGger:A:PULse:WIDth:WHEn

## TRIGger:A:PULse:WIDth:WIDth

This commands sets or returns the width setting for the pulse width trigger. This command is provided for compatibility with other Tektronix instruments. See the [TRIGger:A:PULSEWidth:WIDth](#) command.

**Group** Trigger

**Syntax** TRIGger:A:PULse:WIDth:WIDth

## TRIGger:A:PULSEWIDth? (Query Only)

This query-only command returns the width parameters for the pulse width trigger.

**Group** Trigger

**Syntax** TRIGger:A:PULSEWIDth?

**Examples** TRIGGER:A:PULSEWIDth? might return :TRIGGER:A:WIDTH:LOWLIMIT 2.0000E-9; HIGHLIMIT 2.0000E-9; WHEN WITHIN; POLARITY POSITIVE as the current A width trigger parameters.

## TRIGger:A:PULSEWidth:POLarity

This command sets or returns the polarity for the width trigger.

**Group** Trigger

**Syntax**     `TRIGger:A:PULSEwidth:POLarity {NEGative|POSitive}`  
`TRIGger:A:PULSEwidth:POLarity?`

**Arguments**     `NEGative` specifies a negative pulse.  
`POSitive` specifies a positive pulse.

**Examples**     `TRIGGER:A:PULSEWIDTH:POLARITY NEGATIVE` sets the pulse polarity to negative.  
`TRIGGER:A:PULSEWIDTH:POLARITY?` might return  
`:TRIGGER:A:WIDTH:POLARITY POSITIVE` indicating a positive pulse.

## TRIGger:A:PULSEWidth:SOURce

This command sets or returns the source for the pulse width trigger..

**Group**     Trigger

**Syntax**     `TRIGger:A:PULSEwidth:SOURce {CH1|CH2|CH3|CH4|LINE|EXT}`  
`TRIGger:A:PULSEwidth:SOURce?`

**Arguments**     `CH<x>` specifies one input channel as the A edge trigger source.  
`EXT` specifies an external trigger using the Auxiliary Trigger Input connector located on the rear panel of the instrument.  
`LINE` specifies AC line voltage.

**Examples**     `TRIGGER:A:PULSEWIDTH:SOURCE CH1` sets channel 1 as the pulse width source.  
`TRIGGER:A:PULSEWIDTH:SOURCE?` might return `:TRIGGER:A:PULSEWIDTH:SOURCE CH1` indicating that channel 1 is the pulse width source.

## TRIGger:A:PULSEWidth:WHEn

This command sets or returns whether to trigger on a pulse width that falls outside (or within) the specified range of limits.

<b>Group</b>	Trigger
<b>Syntax</b>	<pre>TRIGger:A:PULSEwidth:WHEN {LESSthan MOREthan EQUAL UNEQUAL} TRIGger:A:PULSEwidth:WHEN?</pre>
<b>Related Commands</b>	<a href="#">TRIGger:A:PULse:WIDth:HIGHLimit</a> , <a href="#">TRIGger:A:PULse:WIDth:LOWLimit</a>
<b>Arguments</b>	<p><b>LESSthan</b> argument sets the instrument to trigger if the a runt pulse is detected with width less than the time set by the <a href="#">TRIGger:A:PULSEwidth:WIDth</a> command.</p> <p><b>MOREthan</b> argument sets the instrument to trigger if the a runt pulse is detected with width more than the time set by the <a href="#">TRIGger:A:PULSEwidth:WIDth</a> command.</p> <p><b>EQUAL</b> argument sets the instrument to trigger when the pattern is true for a time period equal to the time period specified in <a href="#">TRIGger:A:PULSEwidth:WIDth</a> within a <math>\pm 5\%</math> tolerance.</p> <p><b>NOTEQUAL</b> argument sets the instrument to trigger when the pattern is true for a time period greater than or less than (but not equal) the time period specified in <a href="#">TRIGger:A:PULSEwidth:WIDth</a> within a <math>\pm 5\%</math> tolerance.</p>
<b>Examples</b>	<p><code>TRIGGER:A:PULSEWIDTH:WHEN LESSTHAN</code> specifies that the duration of the A pulse will fall within defined high and low limits.</p> <p><code>TRIGGER:A:PULSEWIDTH:WHEN?</code> might return <code>:TRIGGER:A:PULSEWIDTH:WHEN MORETHAN</code> indicating the conditions for generating a width trigger.</p>

## TRIGger:A:PULSEwidth:WIDth

This commands sets or returns the width setting for the pulse width trigger.

<b>Group</b>	Trigger
<b>Syntax</b>	<pre>TRIGger:A:PULSEwidth:WIDth &lt;NR3&gt; TRIGger:A:PULSEwidth:WIDth?</pre>
<b>Arguments</b>	<NR3> specifies the pulse width in seconds.

**Examples**     `TRIGGER:A:PULSEWIDTH:WIDTH 5.0E-6` sets the pulse width to 5  $\mu$ s.  
`TRIGGER:A:PULSEWIDTH:WIDTH?` might return `:TRIGGER:A:PULSEWIDTH:WIDTH 2.0000E-9` indicating that the pulse width is set to 2 ns.

## TRIGger:A:RUNT? (Query Only)

This query-only command returns the current A runt trigger parameters. This command query is equivalent to selecting Runt Setup from the Trig menu and then viewing the current settings.

**Group**     Trigger

**Syntax**     `TRIGger:A:RUNT?`

**Examples**     `TRIGGER:A:RUNT?` might return  
`:TRIGGER:A:PULSE:RUNT:LOGIC:INPUT:CH1 HIGH; CH2 X; CH3 X;`  
`CH4 LOW;:TRIGGER:A:PULSE:RUNT:LOGIC:THRESHOLD:CH11.2000; CH2`  
`1.2000;CH3 1.2000;CH4 1.2000; :TRIGGER:A:PULSE:RUNT:POLARITY`  
`POSITIVE :THRESHOLD:HIGH 1.2000; LOW`  
`800.0000E-3;:TRIGGER:A:PULSE:RUNT:WHEN OCCURS;WIDTH`  
`2.0000E-9.`

## TRIGger:A:RUNT:POLarity

This command sets or returns the polarity for the runt trigger. This command is equivalent to selecting Runt Setup from the Trig menu and then choosing the Polarity setting.

**Group**     Trigger

**Syntax**     `TRIGger:A:RUNT:POLarity {EITHer|NEGative|POSitive}`  
`TRIGger:A:RUNT:POLarity?`

**Arguments**     `POSitive` indicates that the rising edge crosses the low threshold and the falling edge re-crosses the low threshold without either edge ever crossing the high threshold.  
`NEGative` indicates that the falling edge crosses the high threshold and the rising edge re-crosses the high threshold without either edge ever crossing the low threshold.



EITHer triggers on a runt of either polarity.

- Examples** TRIGGER:A:RUNT:POLARITY NEGATIVE specifies that the polarity of the A pulse runt trigger is negative.
- TRIGGER:A:RUNT:POLARITY? might return :TRIGGER:A:RUNT:POLARITY POSITIVE indicating that the polarity of the A pulse runt trigger is positive.

## TRIGger:A:RUNT:SOURce

This command sets or returns the source for the A runt trigger.

- Group** Trigger
- Syntax** TRIGger:A:RUNT:SOURce {CH1|CH2|CH3|CH4}  
TRIGger:A:RUNT:SOURce?
- Arguments** CH<x> specifies one of the input channels, which range from 1 through 4 for four-channel instruments and 1 through 2 for two-channel instruments.
- Examples** TRIGGER:A:RUNT:SOURCE CH4 sets channel 4 as the source for the A pulse trigger.
- TRIGGER:A:RUNT:SOURCE? might return :TRIGGER:A:RUNT:SOURCE CH2 indicating that channel 2 is the source for the A pulse trigger.

## TRIGger:A:RUNT:WHEn

This command sets or returns the type of pulse width the trigger checks for when it detects a runt.

- Group** Trigger
- Syntax** TRIGger:A:RUNT:WHEn {LESSthan|MOREthan|EQua1|UNEQua1|OCCURS}  
TRIGger:A:RUNT:WHEn?
- Arguments** OCCURS argument specifies a trigger event if a runt of any detectable width occurs.
- LESSthan argument sets the instrument to trigger if the a runt pulse is detected with width less than the time set by the [TRIGger:A:PULse:RUNT:WIDth](#) command.

**MOREthan** argument sets the instrument to trigger if the a runt pulse is detected with width more than the time set by the **TRIGger:A:PULse:RUNT:WIDth** command.

**EQUa1** argument sets the instrument to trigger when the pattern is true for a time period equal to the time period specified in **TRIGger:A:PULse:RUNT:WIDth** within a  $\pm 5\%$  tolerance.

**NOTEQUa1** argument sets the instrument to trigger when the pattern is true for a time period greater than or less than (but not equal) the time period specified in **TRIGger:A:PULse:RUNT:WIDth** within a  $\pm 5\%$  tolerance.

### Examples

**TRIGGER:A:RUNT:WHEN MORETHAN** sets the runt trigger to occur when the instrument detects a runt in a pulse wider than the specified width.

**TRIGGER:A:RUNT:WHEN?** might return **:TRIGGER:A:PULSE:RUNT:WHEN OCCURS** indicating that a runt trigger will occur if the instrument detects a runt of any detectable width.

## TRIGger:A:RUNT:WIDth

This command sets or returns the minimum width for a runt trigger.

**Group** Trigger

**Syntax** **TRIGger:A:RUNT:WIDth <NR3>**  
**TRIGger:A:RUNT:WIDth?**

**Arguments** **<NR3>** specifies the minimum width, in seconds.

### Examples

**TRIGGER:A:RUNT:WIDTH 15E-6** sets the minimum width of the pulse runt trigger to 15  $\mu$ s.

**TRIGGER:A:RUNT:WIDTH?** might return **:TRIGGER:A:PULSE:RUNT:WIDTH 2.0000E-09** indicating that the minimum width of a pulse runt trigger is 2 ns.

## TRIGger:A:SETHold? (Query Only)

This query-only command returns the clock edge polarity, voltage threshold and source input; data voltage threshold and source; and both setup and hold times for setup and hold violation triggering.

**Group** Trigger

**Syntax** TRIGger:A:SETHold?

**Examples** TRIGGER:A:SETHOLD? might return the settings  
 :TRIGGER:A:LOGIC:SETHOLD:CLOCK:EDGE RISE; THRESHOLD  
 1.4000;SOURCE CH2; :TRIGGER:A:LOGIC:SETHOLD:DATA:THRESHOLD  
 1.4000;SOURCE CH1;:TRIGGER:A:LOGIC:SETHOLD :HOLDTIME  
 2.0000E-9;SETTIME 3.0000E-9

## TRIGger:A:SETHold:CLOCK? (Query Only)

This query-only command returns the clock edge polarity, voltage threshold and source input for setup and hold triggering.

**Group** Trigger

**Syntax** TRIGger:A:SETHold:CLOCK?

**Related Commands** [TRIGger:A:SETHold:DATA?](#), [TRIGger:A:LOGic:SETHold:CLOCK:EDGE](#),  
[TRIGger:A:LOGic:SETHold:CLOCK:THReshold](#), [TRIGger:A:LOGic:SETHold:CLOCK:SOURce](#)

**Examples** TRIGGER:A:SETHOLD:CLOCK? might return  
 :TRIGGER:A:LOGic:SETHold:CLOCK:EDGE RISE;  
 THRESHOLD 1.4000;SOURCE CH2 indicating the current clock settings for  
 setup and hold triggering.

## TRIGger:A:SETHold:CLOCK:EDGE

This command sets or returns the clock edge polarity for setup and hold triggering.

**Group** Trigger

**Syntax** TRIGger:A:SETHold:CLOCK:EDGE {FALL|RISe}  
 TRIGger:A:SETHold:CLOCK:EDGE?

**Arguments** FALL specifies polarity as the clock falling edge.  
 RISe specifies polarity as the clock rising edge.

- Examples**     `TRIGGER:A:SETHOLD:CLOCK:EDGE RISE` specifies the polarity as the clock rising edge.
- `TRIGGER:A:SETHOLD:CLOCK:EDGE?` might return `:TRIGGER:A:SETHOLD:CLOCK:EDGE RISE` indicating that polarity is specified as the clock rising edge.

## TRIGger:A:SETHold:CLOCK:SOURce

This command sets or returns the clock source for the A logic trigger setup and hold input.

- Group**     Trigger
- Syntax**     `TRIGger:A:SETHold:CLOCK:SOURce {CH1|CH2|CH3|CH4|EXT}`  
`TRIGger:A:SETHold:CLOCK:SOURce?`
- Arguments**     `CH<x>` specifies the input channel, which ranges from 1 through 4 for four-channel instruments or 1 through 2 for two channel instruments.
- `EXT` specifies an external trigger using the Auxiliary Trigger Input connector located on the rear panel of the instrument.
- Examples**     `TRIGGER:A:SETHOLD:CLOCK:SOURCE CH1` specifies channel 1 as the A logic setup and hold input.
- `TRIGGER:A:SETHOLD:CLOCK:SOURCE?` might return `:TRIGGER:A:SETHOLD:CLOCK:SOURCE CH4` indicating that channel 4 is the clock source for the setup and hold input. For the A logic trigger.

## TRIGger:A:SETHold:CLOCK:THReshold

This command sets or returns the clock voltage threshold for the setup and hold trigger.

- Group**     Trigger
- Syntax**     `TRIGger:A:SETHold:CLOCK:THReshold {<NR3>|ECL|TTL}`  
`TRIGger:A:SETHold:CLOCK:THReshold?`
- Arguments**     `ECL` specifies a preset ECL high level of -1.3 V.
- `TTL` specifies a preset TTL high level of 1.4 V.

<NR3> is the clock level, in volts.

**Examples**     `TRIGGER:A:SETHOLD:CLOCK:THRESHOLD TTL` specifies the preset TTL value of 1.4 V as the clock threshold for the setup and hold trigger.

`TRIGGER:A:SETHOLD:CLOCK:THRESHOLD?` might return  
`:TRIGGER:A:LOGIC:SETHOLD:CLOCK:THRESHOLD 1.2000E+00`  
 indicating that the clock threshold for the setup and hold trigger is 1.2 V.

## TRIGger:A:SETHold:DATa? (Query Only)

This query-only command returns the voltage threshold and data source for the setup and hold trigger. This command is equivalent to selecting Setup/Hold Setup from the Trig menu and then viewing the current data setups.

**Group**     Trigger

**Syntax**     `TRIGger:A:SETHold:DATa?`

**Related Commands**     [TRIGger:A:SETHold:CLOCK?](#)

**Examples**     `TRIGGER:A:SETHOLD:DATA?` might return  
`:TRIGGER:A:SETHOLD:DATA:THRESHOLD 1.4000;SOURCE CH1` indicating the current trigger data settings.

## TRIGger:A:SETHold:DATa:SOURce

This command sets or returns the data source for the setup and hold trigger.

**Group**     Trigger

**Syntax**     `TRIGger:A:SETHold:DATa:SOURce {CH1|CH2|CH3|CH4|EXT}`  
`TRIGger:A:SETHold:DATa:SOURce?`

**Arguments**     `CH<x>` specifies the input channel, which ranges from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments.

`EXT` specifies an external trigger using the Auxiliary Trigger Input connector located on the rear panel of the instrument.

**Examples**     `TRIGGER:A:SETHOLD:DATA:SOURCE CH1` sets channel 1 as the clock source for the setup and hold trigger.

`TRIGGER:A:SETHOLD:DATA:SOURCE?` might return  
`:TRIGGER:A:LOGIC:SETHOLD:DATA:SOURCE CH2` indicating that channel 2 is the current clock source for the setup and hold trigger.

## TRIGger:A:SETHold:DATA:THReshold

This command sets or returns the data voltage threshold for setup and hold trigger.

**Group**     Trigger

**Syntax**     `TRIGger:A:SETHold:DATA:THReshold {<NR3>|ECL|TTL}`  
`TRIGger:A:SETHold:DATA:THReshold?`

**Arguments**     ECL specifies the preset ECL high level of -1.3 V.  
TTL specifies the preset TTL high level of 1.4 V.  
<NR3> is the setup and hold data level, in V.

**Examples**     `TRIGGER:A:SETHOLD:DATA:THRESHOLD TTL` specifies the preset ECL high level of 1.4 V as the current data voltage level for the setup and hold trigger.

`TRIGGER:A:SETHOLD:DATA:THRESHOLD?` might return  
`:TRIGGER:A:SETHOLD:DATA:THRESHOLD 1.2000E+00` indicating that 1.2 V is the current data voltage level for the setup and hold trigger.

## TRIGger:A:SETHold:HOLDTime

This command sets or returns the hold time for setup and hold violation triggering.

**Group**     Trigger

**Syntax**     `TRIGger:A:SETHold:HOLDTime <NR3>`  
`TRIGger:A:SETHold:HOLDTime?`

**Arguments**     <NR3> specifies the hold time setting in seconds. Positive values for hold time occur after the clock edge. Negative values occur before the clock edge.

**Examples** TRIGGER:A:SETHOLD:HOLDTIME 3.0E-3 sets the hold time for the setup and hold trigger to 3 ms.

TRIGGER:A:SETHOLD:HOLDTIME? might return  
:TRIGGER:A:SETHOLD:HOLDTIME 2.0000E-09 indicating that the current hold time for the setup and hold trigger is 2 ns.

## TRIGger:A:SETHold:SETTime

This command sets or returns the setup time for setup and hold violation triggering.

**Group** Trigger

**Syntax** TRIGger:A:SETHold:SETTime <NR3>  
TRIGger:A:SETHold:SETTime?

**Arguments** <NR3> specifies the setup time for setup and hold violation triggering.

**Examples** TRIGGER:A:SETHOLD:SETTIME 3.0E-6 specifies that the current setup time for setup and hold trigger is 3  $\mu$ s.

TRIGGER:A:SETHOLD:SETTIME? might return  
:TRIGGER:A:LOGIC:SETHOLD:SETTIME 2.0000E-09  
indicating that the current setup time for setup and hold trigger is 2 ns.

## TRIGger:A{:TRANSition|:RISEFall}? (Query Only)

This query-only command returns transition time trigger parameters. This command is equivalent to selecting Transition Setup from the Trig menu and then viewing the current transition settings.

**Group** Trigger

**Syntax** TRIGger:A{:TRANSition|:RISEFall}?

**Examples** TRIGGER:A::TRANSITION? might return  
:TRIGGER:A:TRANSITION:DELTATIME 2.0000E-9;POLARITY  
POSITIVE;THRESHOLD:HIGH 1.2000;LOW  
800.0000E-3;:TRIGGER:A:PULSE:TRANSITION:WHEN SLOWERTHAN  
indicating the current transition time trigger settings.

## TRIGger:A{:TRANSition|:RISEFall}:DELTatime

This command sets or returns the delta time used in calculating the transition value for the transition trigger. This is equivalent to selecting Transition Setup from the Trig menu and setting the Time.

<b>Group</b>	Trigger
<b>Syntax</b>	<pre>TRIGger:A{:TRANSition :RISEFall}:DELTatime &lt;NR3&gt; TRIGger:A{:TRANSition :RISEFall}:DELTatime?</pre>
<b>Arguments</b>	<NR3> specifies the delta time, in seconds.
<b>Examples</b>	<p>TRIGGER:A:TRANSITION:DELTATIME 15E-6 sets the delta time of the transition trigger to 15 <math>\mu</math>s.</p> <p>TRIGGER:A:TRANSITION:DELTATIME? might return :TRIGGER:A:TRANSITION:DELTATIME 2.0000E-09 indicating that the delta time of the transition trigger is set to 2 ns.</p>

## TRIGger:A{:TRANSition|:RISEFall}:POLarity

This command sets or returns the polarity for the transition trigger.

<b>Group</b>	Trigger
<b>Syntax</b>	<pre>TRIGger:A{:TRANSition :RISEFall}:POLarity {EITHer NEGative POSitive} TRIGger:A{:TRANSition :RISEFall}:POLarity?</pre>
<b>Arguments</b>	<p>POSitive indicates that a pulse edge must traverse from the lower (most negative) to higher (most positive) level for transition triggering to occur.</p> <p>NEGative indicates that a pulse edge must traverse from the upper (most positive) to lower (most negative) level for transition triggering to occur.</p> <p>EITHer indicates either positive or negative polarity.</p>
<b>Examples</b>	<p>TRIGGER:A:TRANSITION:POLARITY NEGATIVE sets the transition polarity to negative.</p>



TRIGGER:A:TRANSITION:POLARITY? might return :TRIGGER:A:TRANSITION:POLARITY EITHER indicating that the polarity can be either positive or negative.

## TRIGger:A{:TRANSition|:RISEFall}:SOURce

This command sets or returns the source for transition trigger.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:A{:TRANSition :RISEFall}:SOURce {CH1 CH2 CH3 CH4} TRIGger:A{:TRANSition :RISEFall}:SOURce?
<b>Arguments</b>	CH<x> specifies one of the input channels, which range from 1 through 4 for four-channel instruments and 1 through 2 for two-channel instruments.
<b>Examples</b>	TRIGGER:A:TRANSITION:SOURCE CH4 sets channel 4 as the source for the transition trigger.  TRIGGER:A:TRANSITION:SOURCE? might return :TRIGGER:A:TRANSITION:SOURCE CH2 indicating that channel 2 is the source for the A transition trigger.

## TRIGger:A{:TRANSition|:RISEFall}:WHEn

This command sets or returns whether to check for a transitioning signal that is faster or slower than the specified delta time.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:A{:TRANSition :RISEFall}:WHEn {SLOWer FASTer Equal UNEQual} TRIGger:A{:TRANSition :RISEFall}:WHEn?
<b>Arguments</b>	FASTer sets the trigger to occur when the transitioning signal is faster than the set volts/second rate.  SLOWer sets the trigger to occur when the transitioning signal is slower than the set volts/second rate.  Equal sets the trigger to occur when the transitioning signal is equal to the set volts/second rate within a $\pm 5\%$ tolerance.

UNEQUAL sets the trigger to occur when the transitioning signal is not equal to the set volts/second rate  $\pm 5\%$ .

**Examples** TRIGGER:A:TRANSITION:WHEN SLOWERTHAN sets the trigger to occur when the transitioning signal is slower than the set volts/second rate.

TRIGGER:A:TRANSITION:WHEN? might return :TRIGGER:A:TRANSITION:WHEN FASTER THAN indicating that the transition triggers when the transitioning signal is faster than the set volts/second rate.

## TRIGger:A:TYPE

This command sets or returns the type of A trigger. The five types of triggers are of Edge, Logic, Pulse, Serial, and Video. Logic and Pulse triggers contain classes. Logic triggers consist of State, Pattern, and SetHold classes; Pulse triggers consist of Runt, Width, and Transition. Once you have set the trigger type, you may also need to identify the associated trigger class. For details on selecting Logic and Pulse trigger classes, see [TRIGger:A:LOGic:CLass](#) and [TRIGger:A:PULse:CLass](#) respectively. This command is similar to selecting A Event (Main) Trigger Setup from the Trig menu and then selecting the desired Trigger Type.

**Group** Trigger

**Syntax** TRIGger:A:TYPE {EDGE|LOGic|PULse|SERial|VIDeo}  
TRIGger:A:TYPE?

**Related Commands** [TRIGger:A:EDGE?](#), [TRIGger:A:LOGic:CLass](#), [TRIGger:A:PULse:CLass](#)

**Arguments** EDGE is a normal trigger. A trigger event occurs when a signal passes through a specified voltage level in a specified direction and is controlled by the [TRIGger:A:EDGE?](#) commands.

LOGic specifies that a trigger occurs when specified conditions are met and is controlled by the [TRIGger:A:LOGic?](#) commands.

PULse specifies that a trigger occurs when a specified pulse is found and is controlled by the [TRIGger:A:PULse?](#) commands.

SERial specifies that a trigger occurs when a communications signal is found. Supports AMI, HDB3, B3ZS, B6ZS, B8ZS, CMI, MLT3, Manchester, and NRZ encoded communications signals.

VIDeo specifies that the trigger occurs when a video signal is found.

- Examples** `TRIGGER:A:TYPE EDGE` sets the A trigger type to EDGE.
- `TRIGGER:A:TYPE?` might return `:TRIGGER:A:TYPE PULSE` indicating that the A trigger type is a pulse trigger.

## TRIGger:A:UPPerthreshold:CH<x>

This command sets the upper threshold for the channel selected. Each channel can have an independent level. Used for the following trigger types: runt, slew rate.

- Group** Trigger
- Syntax** `TRIGger:A:UPPerthreshold:CH<x> {<NR3>|ECL|TTL}`  
`TRIGger:A:UPPerthreshold:CH<x>?`
- Arguments** ECL specifies a preset ECL high level of -1.3 V.  
 TTL specifies a preset TTL high level of 1.4 V.  
 <NR3> is the clock level, in volts.
- Examples** `TRIGGER:A:UPPERTHRESHOLD:CH2 50E-3` sets the upper limit of the pulse runt trigger to 50 mV for channel 2.
- `TRIGGER:A:UPPERTHRESHOLD:CH2?` might return `:TRIGGER:A:UPPERTHRESHOLD:CH2 1.2000E-01` indicating that the upper limit of the pulse runt trigger is set to 120 mV.

## TRIGger:A:VIDeo? (Query Only)

This query-only command returns the A trigger video parameters.

- Conditions** This command requires a DPO4VID application module.
- Group** Trigger
- Syntax** `TRIGger:A:VIDeo?`
- Examples** `TRIGGER:A:VIDEO?` might return `:TRIGGER:A:VIDEO:CUSTOM:FORMAT INTERLACED; SCAN RATE1; :TRIGGER:A:VIDEO:FIELD ALLFIELDS;`

HOLDOFF:FIELD 1.0000;TIME 20.0000E-3; :TRIGGER:A:VIDEO:LINE  
1;SCAN RATE1; SOURCE CH1;STANDARD NTSC.

## TRIGger:A:VIDeo:CUSTom? (Query Only)

This query-only command returns the A trigger custom video parameters.

<b>Conditions</b>	This command requires a DPO4VID application module.
<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:A:VIDeo:CUSTom?
<b>Examples</b>	TRIGGER:A:VIDEO:CUSTOM? might return the parameters :TRIGGER:A:VIDEO:CUSTOM:FORMAT INTERLACED; SCAN RATE1.

## TRIGger:A:VIDeo:CUSTom{:FORMat[:TYPE]}

This command sets or returns the video trigger format. Use this command only when the video format is set to custom.

<b>Conditions</b>	This command requires a DPO4VID application module.
<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:A:VIDeo:CUSTom{:FORMat[:TYPE]} {INTERLAcED PROGressive} TRIGger:A:VIDeo:CUSTom{:FORMat[:TYPE]}?
<b>Arguments</b>	INTERLACED argument sets the format for interlaced video lines.  PROGressive argument sets the format for progressive video lines.
<b>Examples</b>	TRIGGER:A:VIDEO:CUSTOM:FORMAT PROGRESSIVE sets the custom format for the A video trigger to progressive lines.  TRIGGER:A:VIDEO:CUSTOM:FORMAT? might return :TRIGGER:A:VIDEO:CUSTOM:FORMAT INTERLACED indicating that interlaced is selected as the custom format for the A video trigger.

## TRIGger:A:VIDeo:CUSTom:SCAN

This command sets or returns the horizontal line scan rate of the A video trigger. Use this command only when the video format is set to custom. This is equivalent to selecting Video Setup from the Trig menu, setting Custom for Format, and selecting the Scan Rate from the drop-down menu.

**Conditions** This command requires a DPO4VID application module.

**Group** Trigger

**Syntax** TRIGger:A:VIDeo:CUSTom:SCAN  
{RATE15K|RATE20K|RATE25K|RATE35K|RATE50K}  
TRIGger:A:VIDeo:CUSTom:SCAN?

**Arguments** RATE15 argument sets the range of the video line scan rate to 15 kHz through 20 kHz. This is the standard broadcast rate.

RATE20 argument sets the range of the video line scan rate to 20 kHz through 25 kHz.

RATE25 argument sets the range of the video line scan rate to 25 kHz through 35 kHz

RATE35 argument sets the range of the video line scan rate to 35 kHz through 50 kHz

RATE50 argument sets the range of the video line scan rate to 50 kHz through 65 kHz

**Examples** TRIGGER:A:VIDEO:CUSTOM:SCAN RATE15 sets the scan rate of the A trigger custom video to Rate 1, which is 15 kHz to 20 kHz (standard broadcast rate).

TRIGGER:A:VIDEO:CUSTOM:SCAN? might return :TRIGGER:A:VIDEO:CUSTOM:SCAN RATE20 indicating that the video line rate for the A trigger custom video is set to Rate20, which is 20 kHz to 25 kHz.

## TRIGger:A:VIDeo:HDtv:FORMat

Sets or queries the analog HDTV video signal format on which to trigger.

**Conditions** This command requires a DPO4VID application module.

**Group** Trigger

**Syntax** TRIGger:A:VIDeo:HDtv:FORMat  
{HD1080P24|HD720P60|HD480P60|HD1080I50|HD1080P25|HD1080I60|HD1080PSF24}  
TRIGger:A:VIDeo:HDtv:FORMat?

**Arguments** Table 2-37: Available HDTV formats

HDTV format	Description
1080I50	1125 Lines (1080 active), 1920 x 1080 pixel, interlaced, 60 fps
1080I60	1125 lines (1080 active), 1920 x 1080 pixel, interlaced, 50 fps
1080P24	1125 lines (1080 active), 1920 x 1080 pixel, progressive, 24 fps
1080P25	1125 lines (1080 active), 1920 x 1080 pixel, progressive, 25 fps
1080SF24	1125 Lines (1080 active), 1920 x 1080 pixel, progressive (sF), 24 fps
720P60	750 lines (720 active), 1280 x 720 pixel, progressive, 60 fps
480P60	525 lines (480 active), 640 or 704 x 480 pixel, progressive, 60 fps

## TRIGger:A:VIDeo:HOLDoff:FIELD

This command sets or returns the video trigger holdoff in terms of video fields.

**Conditions** This command requires a DPO4VID application module.

**Group** Trigger

**Syntax** TRIGger:A:VIDeo:HOLDoff:FIELD <NR3>  
TRIGger:A:VIDeo:HOLDoff:FIELD?

**Arguments** <NR3> argument is a real number from 0.0 to 8.5 in increments of 0.5. The argument sets the number of fields that the instrument waits before rearming the video trigger.

- Examples** `TRIGGER:A:VIDEO:HOLDOFF:FIELD?` might return  
`:TRIGger:A:VIDeo:HOLdoFF:FIELD 5` indicating that the instrument is set  
to wait 5 video fields before rearming the trigger.
- `TRIGGER:A:VIDEO:HOLDOFF:FIELD 4.5` sets the instrument to wait 4.5 video  
fields before rearming the trigger.

## TRIGger:A:VIDeo:LINE

This command sets or returns the video line number on which the instrument  
triggers. This command is equivalent to selecting Video Setup from the Trig  
menu, selecting Line # in the Trigger on box, and setting the line number. Use the  
[TRIGger:A:VIDeo{:SYNC|:FIELD}](#) command to actually trigger the instrument  
on the line that you specify with this command.

- Conditions** This command requires a DPO4VID application module.

- Group** Trigger

- Syntax** `TRIGger:A:VIDeo:LINE <NR1>`  
`TRIGger:A:VIDeo:LINE?`

- Related Commands** [TRIGger:A:VIDeo{:SYNC|:FIELD}](#)

- Arguments** `<NR1>` argument is an integer that sets the video line number on which the  
instrument triggers. The following table lists the valid choices, depending on the  
active video standard.

**Table 2-38: Video Line Numbering Ranges**

Video Standard	Line Number Range
CUSTOM	4–3000
NTSC	1–263 (odd) and 264–525 (even)
PAL	1–625
SECAM	1–625
HD480P60	1–520
HD720P60	1–750
HD1080150	1–1125
HD1080160	1–1125
HD1080P24	1–1125

**Table 2-38: Video Line Numbering Ranges, (cont.)**

Video Standard	Line Number Range
HD1080P25	1–1125
HD1080SF24	1–1125

**Examples**     `TRIGGER:A:VIDEO:LINE 23` sets the instrument to trigger on the line 23.

`TRIGGER:A:VIDEO:LINE?` might return `:TRIGGER:A:VIDEO:LINE 10` indicating that the instrument is set to trigger on line 10.

## TRIGGER:A:VIDEO:POLARITY

This command sets or returns the polarity of the A video trigger.

**Conditions**     This command requires a DPO4VID application module.

**Group**     Trigger

**Syntax**     `TRIGGER:A:VIDEO:POLARITY {NEGATIVE|POSITIVE}`  
`TRIGGER:A:VIDEO:POLARITY?`

**Arguments**     `POSITIVE` argument sets the instrument to trigger on a positive video sync pulse.  
`NEGATIVE` argument sets the instrument to trigger on a negative video sync pulse.

**Examples**     `TRIGGER:A:VIDEO:POLARITY NEGATIVE` sets the instrument to trigger on a negative video pulse.

`TRIGGER:A:VIDEO:POLARITY?` might return `:TRIGGER:A:VIDEO:POLARITY POSITIVE` indicating that the instrument is set to trigger on a positive video sync pulse.

## TRIGGER:A:VIDEO:SOURCE

This command sets or returns the source for the A video trigger. This command is equivalent to selecting Video Setup from the Trig menu and selecting a channel from the Source drop-down menu.

**Conditions**     This command requires a DPO4VID application module.



<b>Group</b>	Trigger
<b>Syntax</b>	<pre>TRIGger:A:VIDeo:SOURce {CH&lt;x&gt;} TRIGger:A:VIDeo:SOURce?</pre>
<b>Arguments</b>	CH<x> argument specifies one of the input channels of the instrument as the A video trigger. The value of x ranges from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments.
<b>Examples</b>	<p>TRIGGER:A:VIDEO:SOURCE CH1 sets the source for A video trigger to Channel 1.</p> <p>TRIGGER:A:VIDEO:SOURCE? might return :TRIGger:A:VIDeo:SOURce CH2 indicating that the source for the A video trigger is set to Channel 2.</p>

## TRIGger:A:VIDeo:STANdard

This command sets or returns the video standard.

**Conditions** This command requires a DPO4VID application module.

**Group** Trigger

**Syntax**

```
TRIGger:A:VIDeo:STANdard {CUSTom|NTSc|PAL|
SECAM|HDtv|EIA343}
TRIGger:A:VIDeo:STANdard?
```

**Arguments** CUSTom argument sets the instrument to use custom video horizontal scan rate parameters that you set with the [TRIGger:A:VIDeo:CUSTom:SCAN](#) command.

NTSc argument sets the instrument to trigger on video signals that meet the NTSC 525/60/2:1 standard (a line rate of 525 lines per frame and a field rate of 60 Hz).

PAL argument sets the instrument to trigger on video signals that meet the NTSC 625/50/2:1 standard (a line rate of 625 lines per frame and a field rate of 50 Hz).

SECAM argument sets the instrument to trigger on video signals that meet the SECAM standard.

HDtv argument sets the instrument to trigger on an HDTV format.

EIA343 argument sets the instrument to trigger on EIA343 format.

- Examples**     `TRIGGER:A:VIDEO:STANDARD` NTSC sets the instrument to trigger on NTSC-standard video signals.
- `TRIGGER:A:VIDEO:STANDARD?` might return `:TRIGGER:A:VIDEO:STANDARD` HDTV indicating that the instrument is set to trigger on an HDTV format.

## TRIGger:A:VIDeo{:SYNC|:FIELD}

This command sets or returns the video field or line that the trigger detects.

- Conditions**     This command requires a DPO4VID application module.

- Group**     Trigger

- Syntax**     `TRIGger:A:VIDeo{:SYNC|:FIELD}`  
                   `{ODD|EVEN|ALLFieLds|ALLLines|NUMERic}`  
                   `TRIGger:A:VIDeo{:SYNC|:FIELD}?`

- Arguments**     `ODD` argument sets the instrument to trigger on interlaced video odd fields.
- `EVEN` argument sets the instrument to trigger on interlaced video even fields.
- `ALLFieLds` argument sets the instrument to trigger on all fields.
- `ALLLines` argument sets the instrument to trigger on all video lines.
- `NUMERic` argument sets the instrument to trigger on the video signal line specified by the `TRIGger:A:VIDeo:LINE` command.

- Examples**     `TRIGGER:A:VIDEO:FIELD EVEN` sets the A video trigger so that it will trigger on even fields.
- `TRIGGER:A:VIDEO:FIELD?` might return `:TRIGGER:A:VIDEO:FIELD` `ALLFIELDS` indicating that the A video will trigger on all video fields.

## TRIGger:B

This command sets the B trigger level to 50% of minimum and maximum. The query form of this command returns the B trigger parameters. This command is similar to selecting B Event (Delayed) Trigger Setup from the Trig menu and then viewing the current setups.

- Group**     Trigger

<b>Syntax</b>	TRIGger:B SETLeve1 TRIGger:B?
<b>Related Commands</b>	<a href="#">TRIGger:A</a>
<b>Arguments</b>	SETLeve1 sets the B trigger level to 50% of MIN and MAX.
<b>Examples</b>	TRIGGER:B SETLEVEL sets the B trigger level to 50% of MIN and MAX.  TRIGGER:B? might return the following B trigger parameters: :TRIGGER:B:STATE 0;TYPE EDGE; LEVEL -220.0000E-3;BY TIME;EDGE:SOURCE CH1;SLOPE RISE;COUPLING DC; :TRIGGER:B:TIME 16.0000E-9;EVENTS:COUNT 2

## TRIGger:B:BY

This command selects or returns whether the B trigger occurs after a specified number of events or a specified period of time after the A trigger. This is equivalent to selecting B Event (Delayed) Trigger Setup from the Trig menu, selecting the A →B Seq tab, and then choosing Trig After Time or Trig on nth event.

---

**NOTE.** *The traditional Runs After functionality is now served by the Horizontal Delay function. For details, see the [HORizontal\[:MAIn\]:DElay:MODE](#) and [HORizontal\[:MAIn\]:DElay:TIME](#) commands.*

---

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:B:BY {EVENTS TIME} TRIGger:B:BY?
<b>Related Commands</b>	<a href="#">TRIGger:B:EVENTS:COUNT</a> , <a href="#">TRIGger:B:TIME</a> , <a href="#">HORizontal[:MAIn]:DElay:MODE</a> , <a href="#">HORizontal[:MAIn]:DElay:TIME</a>
<b>Arguments</b>	EVENTS sets the B trigger to take place following a set number of trigger events after the A trigger occurs. The number of events is specified by TRIGger:B:EVENTS:COUNT.  TIME sets the B trigger to occur a set time after the A trigger event. The time period is specified by TRIGger:B:TIME.

**Examples**     `TRIGGER:B:BY TIME` sets the B trigger to occur at a set time after the A trigger event.

`TRIGGER:B:BY?` might return `:TRIGGER:B:BY EVENTS` indicating that the B trigger takes place following a set number of trigger events after the A trigger occurs.

## TRIGger:B:EDGE? (Query Only)

This query-only command returns the source, slope, and coupling for B trigger. This command is equivalent to selecting B Event (Delayed) Trigger Setup from the Trig menu and viewing the current Source, Slope, and Coupling settings.

**Group**     Trigger

**Syntax**     `TRIGger:B:EDGE?`

**Related Commands**     [TRIGger:B:EDGE:COUPling](#), [TRIGger:B:EDGE:SLOpe](#), [TRIGger:B:EDGE:SOUrce](#)

**Examples**     `TRIGGER:B:EDGE?` might return `:TRIGGER:B:EDGE:SOURCE CH1; SLOPE RISE;COUPLING DC`

## TRIGger:B:EDGE:COUPling

This command sets or returns the type of coupling for the B trigger. This command is equivalent to selecting B Event (Delayed) Trigger Setup from the Trig menu and choosing the setting from the Coupling drop-down list.

**Group**     Trigger

**Syntax**     `TRIGger:B:EDGE:COUPling {DC|HFRej|LFRej|NOISerej}`  
`TRIGger:B:EDGE:COUPling?`

**Related Commands**     [TRIGger:B:EDGE?](#)

**Arguments**     DC selects DC trigger coupling.

                     HFRej selects high-frequency reject coupling.

                     LFRej selects low-frequency reject coupling.

NOISErej selects DC low sensitivity.

**Examples** TRIGGER:B:EDGE:COUPLING DC selects DC for the B trigger coupling.  
 TRIGGER:B:EDGE:COUPLING? might return :TRIGGER:B:EDGE:COUPLING  
 ATRIGGER for the B trigger coupling.

## TRIGger:B:EDGE:SLOpe

This command sets or returns the slope for the B trigger. This command is equivalent to selecting B Event (Delayed) Trigger Setup from the Trig menu and choosing the Slope.

**Group** Trigger

**Syntax** TRIGger:B:EDGE:SLOpe {RISe|FALL}  
 TRIGger:B:EDGE:SLOpe?

**Related Commands** [TRIGger:B:EDGE?](#)

**Arguments** RISe argument specifies the trigger on the rising or positive edge of a signal.  
 FALL argument specifies the trigger on the falling or negative edge of a signal.

**Examples** TRIGGER:B:EDGE:SLOPE FALL sets the B edge trigger to occur on the falling slope.  
 TRIGGER:B:EDGE:SLOPE? might return :TRIGGER:B:EDGE:SLOPE RISE indicating that the B edge trigger occurs on the rising slope.

## TRIGger:B:EDGE:SOUrce

This command sets or returns the source for the B trigger. This command is equivalent to selecting B Event (Delayed) Trigger Setup from the Trig menu and choosing the desired setting from the Source drop-down list.

**Group** Trigger

**Syntax** TRIGger:B:EDGE:SOUrce {CH<x>|EXT}  
 TRIGger:B:EDGE:SOUrce?

**Related Commands**    [TRIGger:B:EDGE?](#)

**Arguments**    CH<x> specifies one of the input channels as the B trigger source. Input channels are specified by x, which can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments.

EXT specifies an external trigger (using the Auxiliary Trigger Input connector, located on the rear panel of the instrument) as the B trigger source.

**Examples**    TRIGGER:B:EDGE:SOURCE CH4 sets channel 4 as the input source for the B trigger.

TRIGGER:B:EDGE:SOURCE? might return :TRIGGER:B:EDGE:SOURCE CH1 indicating that the current input source for the B trigger is channel 1.

## TRIGger:B:EVENTS? (Query Only)

This query-only command returns the current B trigger events parameter. This command is equivalent to selecting B Event (Delayed) Trigger Setup from the Trig menu, selecting the A →B Seq tab, choosing Trig on nth event, and viewing the Trig Event setting.

**Group**    Trigger

**Syntax**    TRIGGER:B:EVENTS?

**Related Commands**    [TRIGger:B:EVENTS:COUNt](#)

**Examples**    TRIGGER:B:EVENTS? might return

:TRIGGER:B:EVENTS:COUNt 2

indicating that 2 events must occur before the B trigger occurs.

## TRIGger:B:EVENTS:COUNt

This command sets or returns the number of events that must occur before the B trigger (when TRIG:DElay:BY is set to EVENTS). This command is equivalent to selecting B Event (Delayed) Trigger Setup from the Trig menu, selecting the A →B Seq tab, choosing Trig on nth event, and setting the desired Trig Event value.

**Group**    Trigger

**Syntax**     `TRIGger:B:EVENTS:COUNT <NR1>`  
`TRIGger:B:EVENTS:COUNT?`

**Related Commands**     [TRIGger:B:EVENTS?](#)

**Arguments**     <NR1> is the number of B trigger events, which can range from 1 to 10,000,000.

**Examples**     `TRIGGER:B:EVENTS:COUNT 4` specifies that the B trigger will occur four trigger events after the A trigger.

`TRIGGER:B:EVENTS:COUNT?` might return `:TRIGGER:B:EVENTS:COUNT 2` indicating that two events must occur after the A trigger before the B trigger can occur.

## TRIGger:B:LEVel

This command sets or returns the level for the B trigger. This command is equivalent to selecting B Event (Delayed) Trigger Setup from the Trig menu, selecting the A →B Seq tab and setting the B Trig Level voltage.

**Group**     Trigger

**Syntax**     `TRIGger:B:LEVel {ECL|TTL|<NR3>}`  
`TRIGger:B:LEVel?`

**Related Commands**     [TRIGger:A:LEVel](#), [TRIGger:B](#), [TRIGger:B:EDGE:SOURce](#)

**Arguments**     ECL specifies a preset ECL level of -1.3 V.  
TTL specifies a preset TTL level of 1.4 V.  
<NR3> is the B trigger level, in volts.

**Examples**     `TRIGGER:B:LEVEL ECL` sets the B trigger level to -1.3 V.

`TRIGGER:B:LEVEL?` might return `:TRIGGER:B:LEVEL 173.0000E-03` indicating that the B trigger level is currently set at 173 mV.

## TRIGger:B:LEVel:CH<x>

This command sets or returns the B trigger level for the specified channel. Each Channel can have an independent Level.

<b>Group</b>	Trigger
<b>Syntax</b>	<pre>TRIGger:B:LEVel:CH&lt;x&gt; {&lt;NR3&gt; ECL TTL} TRIGger:B:LEVel:CH&lt;x&gt;?</pre>
<b>Arguments</b>	<p>ECL specifies the ECL high level.</p> <p>TTL specifies the TTL high level.</p> <p>&lt;NR3&gt; specifies the trigger level in user units (usually volts).</p>
<b>Examples</b>	<p>TRIGGER:B:LEVEL:CH2? might return :TRIGGER:B:LEVEL:CH2 1.3000E+00 indicating that the B edge trigger is set to 1.3 V for channel 2.</p> <p>TRIGGER:B:LEVEL:CH3 TTL sets the B edge trigger to TTL high level for channel 3.</p>

## TRIGger:B:LOWerthreshold:CH<x>

This command sets or returns the B trigger lower threshold for the channel selected. Each channel can have an independent level.

<b>Group</b>	Trigger
<b>Syntax</b>	<pre>TRIGger:B:LOWerthreshold:CH&lt;x&gt; {&lt;NR3&gt; ECL TTL} TRIGger:B:LOWerthreshold:CH&lt;x&gt;?</pre>
<b>Arguments</b>	<p>ECL specifies a preset ECL high level of -1.3 V.</p> <p>TTL specifies a preset TTL high level of 1.4 V.</p> <p>&lt;NR3&gt; is the clock level, in volts.</p>

## TRIGger:B:STATE

This command sets or returns the state of B trigger activity. If the B trigger state is on, the B trigger is part of the triggering sequence. If the B trigger state is off, then only the A trigger causes the trigger event.



<b>Group</b>	Trigger
<b>Syntax</b>	<pre>TRIGger:B:STATE {ON OFF &lt;NR1&gt;} TRIGger:B:STATE?</pre>
<b>Related Commands</b>	<a href="#">TRIGger:A:MODE</a>
<b>Arguments</b>	<p>ON argument indicates that the B trigger is active and in causes trigger events conjunction with the A trigger.</p> <p>OFF argument indicates that only the A trigger causes trigger events.</p> <p>&lt;NR1&gt;</p> <p>A 0 turns off the B trigger; any other value activates the B trigger.</p>
<b>Examples</b>	<p>TRIGGER:B:STATE ON sets the B trigger to active, making it capable of causing trigger events.</p> <p>TRIGGER:B:STATE? might return :TRIGGER:B:STATE 0 indicating that the B trigger is inactive and that only the A trigger causes trigger events.</p>

## TRIGger:B:TIME

This command sets or returns B trigger delay time. The B Trigger time applies only if TRIGger:B:BY is set to TIME. This command is equivalent to selecting B Event (Delayed) Trigger Setup from the Trig menu, choosing the A®B Seq tab, and setting Trig Delay.

<b>Group</b>	Trigger
<b>Syntax</b>	<pre>TRIGger:B:TIME &lt;NR3&gt; TRIGger:B:TIME?</pre>
<b>Related Commands</b>	<a href="#">TRIGger:B:BY</a> , <a href="#">TRIGger:B:EVENTS:COUNt</a>
<b>Arguments</b>	<NR3> is the B trigger delay time in seconds.
<b>Examples</b>	<p>TRIGGER:B:TIME 4E-6 sets the B trigger delay time to 4 <math>\mu</math>s.</p> <p>TRIGGER:B:TIME? might return :TRIGGER:B:TIME 16.0000E-9 indicating that the B trigger time is set to 16 ns.</p>

## TRIGger:B:TYPE

This command sets or returns the type of B trigger. This command is equivalent to selecting B Event (Delayed) Trigger Setup from the Trig menu and choosing Edge.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:B:TYPE EDGE TRIGger:B:TYPE?
<b>Related Commands</b>	<a href="#">TRIGger:A:TYPE</a>
<b>Arguments</b>	EDGE sets the B trigger type to edge.
<b>Examples</b>	TRIGGER:B:TYPE EDGE sets the B trigger type to edge. TRIGGER:B:TYPE? might return :TRIGGER:B:TYPE EDGE.

## TRIGger:B:UPPerthreshold:CH<x>

This command sets the upper threshold for the channel selected. Each channel can have an independent level. Used for the following trigger types: runt, slew rate

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:B:UPPerthreshold:CH<x> {<NR3> ECL TTL} TRIGger:B:UPPerthreshold:CH<x>?
<b>Arguments</b>	ECL specifies a preset ECL high level of -1.3 V. TTL specifies a preset TTL high level of 1.4 V. <NR3> is the clock level, in volts.

## TRIGger:STATE? (Query Only)

This query-only command returns the current state of the triggering system. This command is equivalent to viewing the trigger status LEDs on the instrument front panel.

**Group** Trigger

**Syntax** TRIGger:STATE?

**Related Commands** [TRIGger:A:MODE](#)

**Returns** ARMED indicates that the instrument is acquiring pretrigger information.  
AUTO indicates that the instrument is in the automatic mode and acquires data even in the absence of a trigger.  
DPO indicates that the instrument is in DPO mode.  
PARTIAL indicates that the A trigger has occurred and the instrument is waiting for the B trigger to occur.  
READY indicates that all pretrigger information has been acquired and that the instrument is ready to accept a trigger.  
SAVE indicates that the instrument is in save mode and is not acquiring data.  
TRIGGER indicates that the instrument triggered and is acquiring the post trigger information.

**Examples** TRIGGER:STATE? might return :TRIGGER:STATE ARMED indicating that the pretrigger data is being acquired.

## \*TST? (Query Only)

This query-only command tests (self-test) the GPIB interface and returns a 0.

**Group** Miscellaneous

**Syntax** \*TST?

**Examples** \*TST? always returns 0.

## UNLock (No Query Form)

This command (no query form) unlocks the front panel. The command is equivalent to LOCK NONE.

---

**NOTE.** *If the instrument is in the Remote With Lockout State (RWLS), the UNLock command has no effect. For more information, see the ANSI-IEEE Std 488.1-1987 Standard Digital Interface for Programmable Instrumentation, section 2.8.3 on RL State Descriptions.*

---

<b>Group</b>	Miscellaneous
<b>Syntax</b>	UNLock ALL
<b>Related Commands</b>	<a href="#">LOCK</a>
<b>Arguments</b>	ALL specifies that all front-panel buttons and knobs are unlocked.
<b>Examples</b>	UNLOCK ALL unlocks all front-panel buttons and knobs.

## VERBoSe

This command sets or returns the Verbose state that controls the length of keywords on query responses. Keywords can be both headers and arguments.

---

**NOTE.** *This command does not affect IEEE Std 488.2-1987 Common Commands (those starting with an asterisk). However, this command does make a corresponding change in the Response Header Enable State of the opposite interface (physical or virtual GPIB interface). Refer to Introduction for more information.*

---

<b>Group</b>	Miscellaneous
<b>Syntax</b>	VERBoSe {OFF ON <NR1>}
<b>Related Commands</b>	<a href="#">HEADer</a> , <a href="#">*LRN?</a> , <a href="#">SET?</a>
<b>Arguments</b>	<p>OFF sets the Verbose state to true, which returns full-length keywords for applicable setting queries.</p> <p>ON sets the Verbose state to false, which returns minimum-length keywords for applicable setting queries.</p> <p>&lt;NR1&gt;</p>

A 0 returns minimum-length keywords for applicable setting queries; any other value returns full-length keywords.

**Examples**     `VERBOSE ON` enables the Verbose state.  
`VERBOSE?` might return `:VERBOSE 0` indicating that the Verbose state is disabled.

## \*WAI (No Query Form)

The \*WAI (Wait) command (no query form) prevents the instrument from executing further commands or queries until all pending commands that generate an OPC message are complete. This command allows you to synchronize the operation of the instrument with your application program. For more information, refer to Synchronization Methods.

**Group**     Status and Error

**Syntax**     \*WAI

**Related Commands**     [BUSY?](#), [\\*OPC](#)

**Examples**     \*WAI prevents the instrument from executing any further commands or queries until all pending commands that generate an OPC message are complete.

## WAVFrm? (Query Only)

This query-only command returns `WFMOutpre?` and `CURVe?` data for the waveform as specified by the `DATA:SOURce` command. This command is equivalent to sending both `WFMOutpre?` and `CURVe?`, with the additional provision that the response to `WAVFrm?` is guaranteed to provide a synchronized preamble and curve.

**Group**     Waveform Transfer

**Syntax**     `WAVFrm?`

**Related Commands**     [CURVe](#), [DATA:SOURce](#), [WFMOutpre?](#)

**Examples**      WAVFRM? might return the waveform data as: :WFMOUTPRE:BIT\_NR 8;BN\_FMT RI;BYT\_NR 1; BYT\_OR MSB;ENCDG ASC;NR\_PT 500;PT\_FMT Y; PT\_ORDER LINEAR;PT\_OFF 0;XINCR 400.0000E-12; XZERO 0.0000;XUNIT "s";YMULT 4.0000E-3; YOFF 0.0000;YZERO 0.0000;YUNIT "V"; WFID "Ch1,DC coupling, 100.0mV/div, 200.0ns/div, 5000 points,Samp1 mode"; :CURVE51,50,51,48,51,48,50,49,51,49,51,48,51,48,51,49,50,49,50,48,49,49

## WFMInpre? (Query Only)

This query-only command returns the waveform formatting specification to be applied to the next incoming CURVe command data.

**Group**      Waveform Transfer

**Syntax**      WFMInpre?

**Related Commands**      [WFMOutpre?](#)

**Examples**      WFMINPRE? might return the waveform formatting as :WFMINPRE:BIT\_NR 8;BN\_FMT RI;BYT\_NR 1; BYT\_OR MSB;ENCDG BIN;NR\_PT 500;PT\_FMT Y; PT\_OFF 0;XINCR 2.0000E-6;XZERO 1.7536E-6; XUNIT "s";YMULT 1.0000E-3;YOFF 0.0000; YZERO 0.0000;YUNIT "V"

## WFMInpre:BIT\_Nr

This command sets or returns the number of bits per binary waveform point for the waveform, as specified by the [DATA:DESTination](#) command. This specification is only meaningful when [WFMInpre:ENCdg](#) is set to BIN.

**Group**      Waveform Transfer

**Syntax**      WFMInpre:BIT\_Nr <NR1>  
WFMInpre:BIT\_Nr?

**Related Commands**      [DATA:DESTination](#), [WFMInpre:ENCdg](#), [WFMInpre:BYT\\_Nr](#),  
[WFMOutpre:BIT\\_Nr](#)

**Arguments**      <NR1> number of bits per data point can be 8, 16 ( RI, RP) or 32 (FP).

- Examples**     `WFMINPRE:BIT_NR 16` sets the number of bits per waveform point to 16, for incoming RI and RP binary format data.
- `WFMINPRE:BIT_NR?` might return `:WFMINPRE:BIT_NR 8` indicating that incoming RI or RP binary format data uses 8 bits per waveform point.

## WFMinpre:BN\_Fmt

This command sets or returns the format of binary data for incoming waveforms.

**Group**     Waveform Transfer

**Syntax**     `WFMinpre:BN_Fmt {RI|RP}`  
`WFMinpre:BN_Fmt?`

**Related Commands**     [WFMinpre:BN\\_Fmt](#)

**Arguments**     RI specifies signed integer data point representation.  
                      RP specifies positive integer data point representation.

**Examples**     `WFMINPRE:BN_FMT FP` specifies that incoming data will be interpreted as single-precision binary floating point numbers.

`WFMINPRE:BN_FMT?` might return `:WFMINPRE:BN_FMT RI` indicating that the incoming data is currently interpreted as signed integers.

## WFMinpre:BYT\_Nr

This command sets or returns the binary field data width for the first ordered waveform, as specified by the [DATA:DESTination](#) command. This specification is only meaningful when [WFMinpre:ENCdg](#) is set to BIN and [WFMinpre:BN\\_Fmt](#) is set to either RI or RP.

**Group**     Waveform Transfer

**Syntax**     `WFMinpre:BYT_Nr <NR1>`  
`WFMinpre:BYT_Nr?`

<b>Related Commands</b>	<a href="#">DATA:DESTination</a> , <a href="#">WFMinpre:BN_Fmt</a> , <a href="#">WFMinpre:ENCdg</a> , <a href="#">WFMinpre:BIT_Nr</a> , <a href="#">WFMinpre:BYT_Nr</a>
<b>Arguments</b>	<NR1> is the number of bytes per data point and can be 1, 2 (RI, RP) or 4 (FP).
<b>Examples</b>	<p><code>WFMINPRE:BYT_NR 1</code> sets the number of bytes per incoming waveform data point to 1, which is the default setting.</p> <p><code>WFMINPRE:BYT_NR?</code> might return <code>:WFMINPRE:BYT_NR 2</code> indicating that there are 2 bytes per incoming waveform data point.</p>

## WFMinpre:BYT\_Or

This command sets or returns which byte of binary waveform data is transmitted first for incoming waveform data when data points require more than one byte. This specification only has meaning when [WFMinpre:ENCdg](#) is set to BIN and [WFMinpre:BYT\\_Nr](#) is greater than 1.

<b>Group</b>	Waveform Transfer
<b>Syntax</b>	<code>WFMinpre:BYT_Or {LSB MSB}</code> <code>WFMinpre:BYT_Or?</code>
<b>Related Commands</b>	<a href="#">WFMinpre:ENCdg</a> , <a href="#">WFMinpre:BYT_Nr</a> , <a href="#">WFMinpre:BYT_Or</a>
<b>Arguments</b>	<p>LSB specifies that the least significant byte will be transmitted first.</p> <p>MSB specifies that the most significant byte will be transmitted first.</p>
<b>Examples</b>	<p><code>WFMINPRE:BYT_OR MSB</code> sets the most significant incoming byte of incoming waveform data to be transmitted first.</p> <p><code>WFMINPRE:BYT_OR?</code> might return <code>:WFMINPRE:BYT_OR LSB</code> indicating that the least significant incoming CURVe data byte will be transmitted first.</p>

## WFMinpre:ENCdg

This command sets or returns the type of encoding for incoming waveform data.

<b>Group</b>	Waveform Transfer
--------------	-------------------



**Syntax**      `WFMInpre:ENCdg {ASC|BIN}`  
`WFMInpre:ENCdg?`

**Related Commands**      [WFMOutpre:ENCdg](#)

**Arguments**      ASC specifies that the incoming data is in ASCII format.

BIN specifies that the incoming data is in a binary format whose further interpretation requires knowledge of BYT\_NR, BIT\_NR, BN\_FMT, and BYT\_OR.

**Examples**      `WFMINPRE:ENCDG ASC` sets the format of incoming waveform data to ASCII format.

`WFMINPRE:ENCDG?` might return `:WFMINPRE:ENCDG BIN` indicating that the incoming waveform data is in binary format.

## WFMInpre:NR\_Pt

This command sets or returns the number of data points that are in the transmitted waveform record.

**Group**      Waveform Transfer

**Syntax**      `WFMInpre:NR_Pt <NR1>`  
`WFMInpre:NR_Pt?`

**Related Commands**      [CURVe](#), [DATa](#), [DATa:STARt](#), [DATa:STOP](#), [SAVe:WAVEform](#),  
[SAVe:WAVEform:FILEFormat](#), [WFMOutpre:NR\\_Pt?](#)

**Arguments**      <NR1> is the number of data points if WFMInpre:PT\_Fmt is set to Y. It is the number of min-max pairs if WFMInpre:PT\_Fmt is set to ENV.

**Examples**      `WFMINPRE:NR_PT 5000` specifies that 5000 data points will be transmitted.

`WFMINPRE:NR_PT?` might return `:WFMINPRE:NR_PT 8000` indicating that there are 8000 data points in the transmitted waveform record.

## WFMinpre:PT\_Fmt

This command sets or returns the point format of the incoming waveform data. Regardless of the argument used, the scale, offset, and so on are interpreted similarly. When ENV is used, waveform data is interpreted over the min-max pair; when Y is used, it is interpreted over a single point.

**Group** Waveform Transfer

**Syntax** WFMinpre:PT\_Fmt {ENV|Y}  
WFMinpre:PT\_Fmt?

**Related Commands** [WFMaxpre:PT\\_Fmt?](#)

**Arguments** ENV specifies that the waveform is transmitted in envelope mode as maximum and minimum point pairs. Only Y values are explicitly transmitted. Absolute coordinates are given by:

$$X_n = XZero + XINcr (n-PT\_Off)$$

$$Y_{nmax} = YZero + YMult (ynmax - YOFf)$$

$$Y_{nmin} = YZero + YMult (ynmin - YOFf)$$

Y specifies a normal waveform where one ASCII or binary data point is transmitted for each point in the waveform record. Only Y values are explicitly transmitted. Absolute coordinates are given by:

$$X_n = XZero + XINcr (N-PT\_Off)$$

$$Y_n = YZero + YMult (Yn - YOFf)$$

**Examples** WFMINPRE:PT\_FMT ENV sets the incoming waveform data point format to enveloped.

WFMINPRE:PT\_FMT? might return :WFMINPRE:PT\_FMT ENV indicating that the waveform is transmitted as maximum and minimum point pairs.

## WFMinpre:PT\_Off

The set form of this command is ignored. The query form always returns a 0 if the waveform specified by [DATA:SOURce](#) is on or displayed. If the waveform is not displayed, the query form generates an error and returns event code 2244. This command is listed for compatibility with other Tektronix oscilloscopes.

**Group** Waveform Transfer

**Syntax** WFMInpre:PT\_Off  
WFMInpre:PT\_Off?

**Related Commands** [DATA:SOURce](#), [WFMOutpre:PT\\_Off?](#)

**Arguments** Arguments are ignored.

## WFMInpre:WFId (No Query Form)

This command (no query form) accepts but ignores the argument. This command is provided only to allow a waveform extracted from the instrument to be easily imported.

**Group** Waveform Transfer

**Syntax** WFMInpre:WFId <QString>

**Related Commands** [DATA:DESTination](#), [WFMInpre:BN\\_Fmt](#), [WFMInpre:ENCdg](#), [WFMOutpre:WFId?](#)

**Arguments** <QString> must be a valid IEEE-488.2 string (but the contents are ignored).

**Examples** WFMINPRE:WFID "CH1, DC COUPLING, 2.000V/DIV, 400.0NS/DIV, 500 POINTS, SAMPLE MODE" is a syntactically correct command.

## WFMInpre:XINcr

This command sets or returns the horizontal interval between incoming waveform points in units specified by WFMInpre:XUNit.

**Group** Waveform Transfer

**Syntax** WFMInpre:XINcr <NR3>  
WFMInpre:XINcr?

**Related Commands**    [WFMinpre:XUNit](#), [WFMOutpre:XINcr?](#)

**Arguments**    <NR3> is the horizontal interval representation.

**Examples**    `WFMINPRE:XINCR 3E-3` sets the interval between Incoming waveform points to 3 ms.

`WFMINPRE:XINCR?` might return `:WFMINPRE:XINCR 1.0000E-3` indicating that if `WFMinpre:XUNit` is set to "s", there is a 1 ms interval between incoming waveform points.

## WFMinpre:XUNit

This command sets or returns the horizontal units of the incoming waveform.

**Group**    Waveform Transfer

**Syntax**    `WFMinpre:XUNit <QString>`  
`WFMinpre:XUNit?`

**Related Commands**    [WFMOutpre:XUNit?](#)

**Arguments**    <QString> contains a maximum of three alpha characters that represent the horizontal unit of measure for the incoming waveform.

**Examples**    `WFMINPRE:XUNIT "HZ"` specifies that the horizontal units for the incoming waveform are hertz.

`WFMINPRE:XUNIT?` might return `:WFMINPRE:XUNIT "s"` indicating that the horizontal units for the incoming waveform are seconds.

## WFMinpre:XZEro

This command sets or returns the sub-sample time between the trigger sample (designated by PT\_OFF) and the occurrence of the actual trigger on the incoming waveform. This value is used to compute TTOFF for the incoming waveform, and is expressed in terms of [WFMinpre:XUNit](#).

**Group**    Waveform Transfer

**Syntax**      `WFMinpre:XZero <NR3>`  
`WFMinpre:XZero?`

**Related Commands**      [WFMinpre:PT\\_Off](#), [WFMinpre:XINcr](#), [WFMinpre:XUNit](#), [WFMinpre:XZero?](#)

**Arguments**      <NR3> argument is a floating point value that ranges from  $-WFMinpre:XINcr$  to 0.

**Examples**      `WFMINPRE:XZERO 5.7E-6` specifies that the trigger actually occurred 5.7  $\mu$ s before the sample designated by [WFMinpre:PT\\_Off](#).  
`WFMINPRE:XZERO?` might return `:WFMINPRE:XZERO 7.5000E-6` indicating that the trigger occurs 7.5  $\mu$ s before the sample designated by [WFMinpre:PT\\_Off](#).

## WFMinpre:YMUlt

This command sets or returns the vertical scale factor (in units/digitizing level) for the reference waveform, specified by [DATA:DESTination](#), upon a [CURVe](#) command.

**Group**      Waveform Transfer

**Syntax**      `WFMinpre:YMUlt <NR3>`  
`WFMinpre:YMUlt?`

**Related Commands**      [DATA:DESTination](#), [WFMinpre:BYT\\_Nr](#), [WFMinpre:YUNit](#)

**Arguments**      <NR3> is the vertical scale factor per digitizing level of the incoming waveform points.

**Examples**      `WFMINPRE:YMULT?` might return `:WFMINPRE:YMULT 40.0000E-3` indicating that the vertical scale is 40 mV/digitizing level (1V/div).  
`WFMINPRE:YMULT 20E-3` specifies that (if `WFMinpre:YUNit` is "V" and `WFMinpre:BYT_Nr` is 1), the vertical scale is 20 mV/digitizing level (500 mV/div).

## WFMinpre:YOff

This command sets or returns the vertical position of the incoming waveform in digitizing levels. Variations in this number are analogous to changing the vertical position of the waveform. For those formats in which [WFMinpre:BYT\\_Nr](#) is

important (all non-floating point formats), this command must take the location of the binary point implied by `BYT_NR` into consideration.

<b>Group</b>	Waveform Transfer
<b>Syntax</b>	<code>WFMInpre:YOFF &lt;NR3&gt;</code> <code>WFMInpre:YOFF?</code>
<b>Related Commands</b>	<a href="#">WFMInpre:BYT_Nr</a> , <a href="#">WFMInpre:YMult</a> , <a href="#">WFMOutpre:YOFF?</a>
<b>Arguments</b>	<code>&lt;NR3&gt;</code> is the vertical offset in digitizing levels.
<b>Examples</b>	<p><code>WFMINPRE:YOFF50</code> specifies that the zero reference point for the incoming waveform is 50 digitizing levels (2 divisions) above the center of the data range.</p> <p><code>WFMINPRE:YOFF?</code> might return <code>:WFMINPRE:YOFF 25</code> indicating the vertical position of the incoming waveform in digitizing levels.</p>

## WFMInpre:YUNit

This command sets or returns the vertical units of the incoming waveform.

<b>Group</b>	Waveform Transfer
<b>Syntax</b>	<code>WFMInpre:YUNit &lt;QString&gt;</code> <code>WFMInpre:YUNit?</code>
<b>Related Commands</b>	<a href="#">WFMOutpre:YUNit?</a>
<b>Arguments</b>	<code>&lt;QString&gt;</code> contains a maximum of three alpha characters that represent the vertical unit of measure for the incoming waveform.
<b>Examples</b>	<p><code>WFMINPRE:YUNIT?</code> might return <code>:WFMINPRE:YUNIT "s"</code> indicating the vertical units for the incoming waveform are seconds.</p> <p><code>WFMINPRE:YUNIT "PA"</code> specifies that the vertical units for the incoming waveform are Pascal.</p>

## WFMinpre:YZero

This command sets or returns the offset of the incoming waveform in units specified by WFMinpre:YUNit. Variations in this number are analogous to changing the vertical offset of the waveform.

**Group** Waveform Transfer

**Syntax** WFMinpre:YZero <NR3>  
WFMinpre:YZero?

**Related Commands** [WFMinpre:YUNit](#), [WFMOutpre:YZero?](#)

**Arguments** <NR3> is the offset in YUNits.

**Examples** WFMINPRE:YZERO 1.5E+0 specifies that the zero reference point for the incoming waveform is 1.5 V below the center of the data range (given that WFMinpre:YUNit is set to V).

WFMINPRE:YZERO? might return :WFMINPRE:YZero 7.5000E-6 indicating that the zero reference for the incoming waveform is 7.5  $\mu$ V below the center of the data range (given that WFMinpre:YUNit is set to V).

## WFMOutpre? (Query Only)

This query-only command returns the waveform formatting data for the waveform specified by the [DATA:SOURce](#) command. The preamble components are considered to be of two types; formatting and interpretation. The formatting components are: ENCDg, BN\_Fmt, BYT\_Or, BYT\_Nr, BIT\_Nr. The interpretation components are derived from the DATA:SOURce specified waveform.

**Group** Waveform Transfer

**Syntax** WFMOutpre?

**Examples** WFMOUTPRE?? might return the waveform formatting data as:  
:WFMOUTPRE:BIT\_NR 8;BN\_FMT RI;BYT\_NR 1; BYT\_OR MSB;ENCDG  
BIN;NR\_PT 500; PT\_FMT Y;PT\_ORDER LINEAR;PT\_OFF 0; XINCR  
8.0000E-9;XZERO 4.8794E-9; XUNIT "s";YMULT -2000.0000E-3;

YOFF -4999.9995E-3;YZERO 0.0000;YUNIT "V"; WFID "Ch1, DC coupling, 2.000V/div, 400.0ns/div,500 points, Sample mode"

## WFMOutpre:BIT\_Nr

This command sets and returns the number of bits per waveform point that outgoing waveforms contain, as specified by the [DATA:SOURce](#) command. Note that values will be constrained according to the underlying waveform data. This specification is only meaningful when [WFMOutpre:ENCdg](#) is set to BIN and [WFMOutpre:BN\\_Fmt](#) is set to either RI or RP. An error is generated if the waveform specified by [DATA:SOURce](#) is not turned on.

**Group** Waveform Transfer

**Syntax** WFMOutpre:BIT\_Nr <NR1>  
WFMOutpre:BIT\_Nr?

**Related Commands** [DATA:SOURce](#), [WFMOutpre:BN\\_Fmt](#), [WFMOutpre:ENCdg](#)

**Arguments** <NR1> number of bits per data point can be 8, 16, 32 or 64.

**Examples** WFMOUTPRE:BIT\_NR 16 sets the number of bits per waveform point to 16 for incoming RI and RP binary format data.

WFMOUTPRE:BIT\_NR? might return :WFMOUTPRE:BIT\_NR 8 indicating that outgoing RI or RP binary format data uses 8 bits per waveform point.

## WFMOutpre:BN\_Fmt

This command sets or returns the format of binary data for outgoing waveforms specified by the [DATA:SOURce](#) command. An error is generated if the waveform specified by [DATA:SOURce](#) is not turned on.

**Group** Waveform Transfer

**Syntax** WFMOutpre:BN\_Fmt {RI|RP}  
WFMOutpre:BN\_Fmt ?

**Related Commands** [DATA:SOURce](#)



<b>Arguments</b>	RI specifies signed integer data point representation.
	RP specifies positive integer data point representation.
<b>Examples</b>	WFMOUTPRE:BN_FMT FP specifies that outgoing waveform data will be in single-precision binary floating point format.
	WFMOUTPRE:BN_FMT? might return :WFMOUTPRE:BN_FMT RI indicating that the outgoing waveform data is currently in signed integer format.

## WFMOutpre:BYT\_Nr

This command sets or returns the binary field data width for the waveform specified by the [DATA:SOURce](#) command. Note that values will be constrained according to the underlying waveform data. This specification is only meaningful when [WFMOutpre:ENCdg](#) is set to BIN, and [WFMOutpre:BN\\_Fmt](#) is set to either RI or RP. An error is generated if the waveform specified by [DATA:SOURce](#) is not turned on.

**Group** Waveform Transfer

**Syntax** WFMOutpre:BYT\_Nr <NR1>  
WFMOutpre:BYT\_Nr?

**Related Commands** [DATA:SOURce](#), [WFMOutpre:BN\\_Fmt](#), [WFMOutpre:ENCdg](#)

**Arguments** <NR1> is the number of bytes per data point and can be 1, 2, 4 or 8. A value of 1 or 2 bytes per waveform point indicates channel data; 4 bytes per waveform point indicate math data; 8 bytes per waveform point indicate pixel map (DPO) data.

**Examples** WFMOUTPRE:BYT\_NR 1 sets the number of bytes per outgoing waveform data point to 1, which is the default setting.

WFMOUTPRE:BYT\_NR? might return :WFMOUTPRE:BYT\_NR 2 indicating that there are 2 bytes per outgoing waveform data point.

## WFMOutpre:BYT\_Or

This command sets or returns which byte of binary waveform data is transmitted first, during a waveform data transfer, when data points require more than one byte. This specification only has meaning when [WFMOutpre:ENCdg](#) is set to BIN. An error is generated if the waveform specified by [DATA:SOURce](#) is not turned on.

<b>Group</b>	Waveform Transfer
<b>Syntax</b>	WFMOutpre:BYT_Or {LSB MSB} WFMOutpre:BYT_Or?
<b>Related Commands</b>	<a href="#">WFMOutpre:ENCdg</a>
<b>Arguments</b>	LSB specifies that the least significant byte will be transmitted first. MSB specifies that the most significant byte will be transmitted first.
<b>Examples</b>	WFMOUTPRE:BYT_OR MSB sets the most significant outgoing byte of waveform data to be transmitted first. WFMOUTPRE:BYT_OR? might return :WFMOUTPRE:BYT_OR LSB indicating that the least significant data byte will be transmitted first.

## WFMOutpre:ENCdg

This command sets and queries the type of encoding for outgoing waveforms. An error is generated if the waveform specified by [DATA:Source](#) is not turned on.

<b>Group</b>	Waveform Transfer
<b>Syntax</b>	WFMOutpre:ENCdg {ASC BIN} WFMOutpre:ENCdg?
<b>Related Commands</b>	<a href="#">DATA:ENCdg</a> , <a href="#">WFMOutpre:BYT_Nr</a> , <a href="#">WFMOutpre:BYT_Or</a> , <a href="#">WFMOutpre:BIT_Nr</a> , <a href="#">WFMOutpre:BN_Fmt</a>
<b>Arguments</b>	ASC specifies that the outgoing data is to be in ASCII format. Waveforms internally stored as integers will be sent as <NR1> numbers, while those stored as floating point will be sent as <NR3> numbers.  BIN specifies that outgoing data is to be in a binary format whose further specification is determined by <a href="#">WFMOutpre:BYT_Nr</a> , <a href="#">WFMOutpre:BIT_Nr</a> , <a href="#">WFMOutpre:BN_Fmt</a> and <a href="#">WFMOutpre:BYT_Or</a> .
<b>Examples</b>	WFMOUTPRE:ENCDG? might return :WFMOUTPRE:ENCDG BIN indicating that outgoing waveform data will be sent in binary format.

WFMOU

```
PRE:ENCDG
```

 ASC specifies that the outgoing waveform data will be sent in ASCII format.

## WFMOU ``` pre:NR_Pt? ``` (Query Only)

This query-only command returns the number of points for the [DATA:SOURce](#) waveform that will be transmitted in response to a [CURVe?](#) query. An error is generated if the waveform specified by [DATA:SOURce](#) is not turned on.

**Group** Waveform Transfer

**Syntax** `WFMOU

```
pre:NR_Pt?
````

**Related Commands** [CURVe](#), [DATA](#), [DATA:STARt](#), [DATA:STOP](#), [SAVe:WAVEform](#), [SAVe:WAVEform:FILEFormat](#), [WFMinpre:NR\\_Pt](#)

**Examples** `WFMOU

```
PRE:NR_PT?
```` might return `:WFMOU

```
PRE:NR_PT
```

 5000` indicating that there are 5000 data points to be sent.

## WFMOU ``` pre:PT_Fmt? ``` (Query Only)

This query-only command returns the point format for the waveform specified by the [DATA:SOURce](#) command. The format specifies a set of equations describing how the scale factors in the preamble are used to give meaning to the [CURVe](#) data points. An error is generated if the waveform specified by [DATA:SOURce](#) is not turned on.

**Group** Waveform Transfer

**Syntax** `WFMOU

```
pre:PT_Fmt?
````

**Related Commands** [CURVe](#), [DATA:SOURce](#)

**Examples** `WFMOU

```
PRE:PT_FMT?
```` might return `:WFMOU

```
pre:PT_Fmt
```

 ENV` indicating that the waveform data is a series of min-max pairs.

## WFMOutpre:PT\_Off? (Query Only)

This query-only command always returns a 0 if the waveform specified by [DATA:SOURce](#) is on or displayed. If the waveform is not displayed, the query form generates an error and returns event code 2244. This command is listed for compatibility with other Tektronix oscilloscopes.

**Group** Waveform Transfer

**Syntax** WFMOutpre:PT\_Off?

**Related Commands** [DATA:SOURce](#)

## WFMOutpre:PT\_ORder? (Query Only)

This query-only command specifies whether the source waveform is Fast Acquisition. A Fast Acquisition waveform is stored as a 200 (vertical) by 500 (horizontal) point bitmap. Each point represents display intensity for that screen location. Only [CURVe?](#) query functions are allowed on Fast Acquisition waveforms.

When the WFMOutpre:PT\_OR query returns Column, this indicates that the source is a Fast Acquisition waveform (and that each of 500 possible horizontal columns being transmitted contains 200 vertical points). When the WFMOutpre:PT\_OR? query returns Linear, this indicates that the source is not a Fast Acquisition waveform (and that each horizontal column being sent contains only one vertical point). Note that waveform points are transmitted in the following order: top to bottom, then left to right. An error is generated if the waveform specified by [DATA:SOURce](#) is not turned on.

**Group** Waveform Transfer

**Syntax** WFMOutpre:PT\_Order?

**Related Commands** [DATA:SOURce](#)

**Examples** WFMOUTPRE:PT\_ORDER? might return :WFMOUTPRE:PT\_ORDER COL specifying that the waveform designated by the DATA:SOURce waveform is a Fast Acquisition waveform.

WFMOUTPRE:PT\_ORDER? might return :WFMOUTPRE:PT\_ORDER LINEAR specifying that the source waveform is a non-Fast Acquisition waveform.

## WFMOutpre:WFId? (Query Only)

This query-only command returns a string describing several aspects of the acquisition parameters for the waveform specified by the [DATA:SOURce](#) command. An error is generated if the waveform specified by [DATA:SOURce](#) is not turned on.

**Group** Waveform Transfer

**Syntax** WFMOutpre:WFId?

**Related Commands** [DATA:SOURce](#)

**Returns** <QString> comprises the following comma-separated fields documented in the tables below:

**Table 2-39: Waveform Suffixes**

| Field         | Description  | Examples   |
|---------------|--|--|
| Source        | The source identification string as it appears in the front-panel scale factor readouts.   | "CH1-4"<br>"Math1-3"<br>"Ref1-4"   |
| Coupling      | A string describing the vertical coupling of the waveform (the Source1 waveform in the case of Dual Waveform Math).  | "AC coupling"<br>"DC coupling"<br>"GND coupling"   |
| Vert Scale    | A string containing the vertical scale factor of the unzoomed waveform. The numeric portion will always be four digits. The examples cover all known internal units.   | "100.0 mV/div"<br>"20.00 dB/div"<br>"45.00 deg/div"<br>"785.4 mrad/div"<br>"500.0 $\mu$ Vs/div"<br>"10.00 kV/s/div"<br>"200.0 mV/div"<br>"50.00 unk/div" |
| Horiz Scale   | A string containing the horizontal scale factor of the unzoomed waveform. The numeric portion will always be four digits. The examples cover all known internal units. | "100 ms/div"<br>"10.00 kHz/div"<br>"50.00 c/div"   |
| Record Length | A string containing the number of waveform points available in the entire record. The numeric portion is given as an integer.  | "500 points"<br>"500000 points"  |

**Table 2-39: Waveform Suffixes, (cont.)**

| Field                    | Description   | Examples  |
|--------------------------|---|---|
| Acquisition Mode         | A string describing the mode used to acquire the waveform.  | "Sample mode"<br>"Pk Detect mode"<br>"Hi Res mode"<br>"Envelope mode"<br>"Average mode" |
| Primary Reference Offset | A string specifying the delta between the Primary Reference (typically the A trigger) and the CURVe? zero reference identified by a combination of PT, Off and XZEro in units of XUNits. For example, in Trigger After Delay, this number would be the actual time between the A and B triggers. In the event that this number is not meaningful, the string will be exactly "0". | "57.2345 ms"<br>"87.3 Hz"<br>"0"  |

**Examples**      WFMOUTPRE:WFID? might return :WFMOUTPRE:WFID "Ch1, DC coupling,100.0mVolts/div,500.0µs/div,500 points, Hi Res mode"

## WFMOutpre:XINcr? (Query Only)

This query-only command returns the horizontal point spacing in units of WFMOutpre:XUNit for the waveform specified by the [DATA:SOURce](#) command. This value corresponds to the sampling interval. An error is generated if the waveform specified by [DATA:SOURce](#) is not turned on.

**Group**      Waveform Transfer

**Syntax**      WFMOutpre:XINcr?

**Related Commands**      [DATA:SOURce](#), [WFMOutpre:XUNit?](#)

**Examples**      WFMOUTPRE:XINCR? might return :WFMOUTPRE:XINCR 10.0000E-6 indicating that the horizontal sampling interval is 10 µs/point (500 µs/div).

## WFMOutpre:XUNit? (Query Only)

This query-only command returns the horizontal units for the waveform specified by the [DATA:SOURce](#) command. An error is generated if the waveform specified by [DATA:SOURce](#) is not turned on.

**Group** Waveform Transfer

**Syntax** WFMOutpre:XUNit?

**Related Commands** [DATA:SOURce](#)

**Examples** WFMOUTPRE:XUNIT? might return :WFMOUTPRE:XUNIT "HZ" indicating that the horizontal units for the waveform are in Hertz.

## WFMOutpre:XZEr0? (Query Only)

This query-only command returns the sub-sample time between the trigger sample (designated by PT\_OFF) and the occurrence of the actual trigger for the waveform specified by the [DATA:SOURce](#) command. This value is in units of WFMOutpre:XUNit. An error is generated if the waveform specified by [DATA:SOURce](#) is not turned on.

---

**NOTE.** During (i.e., ACQuire:STATE OFF), this is the only preamble that changes on each acquisition. If a query is run during steady state operation (that is, all control changes have settled and triggers are arriving on a regular basis), the XZEr0 value of the last stopped state is returned.

---

**Group** Waveform Transfer

**Syntax** WFMOutpre:XZEr0?

**Related Commands** [DATA:SOURce](#), [WFMOutpre:XUNit?](#)

**Examples** WFMOUTPRE:XZEr0? might return :WFMOUTPRE:XZEr0 5.6300E-9 indicating that the trigger actually occurred 5.63 ns before the trigger sample.

## WFMOutpre:YMult? (Query Only)

This query-only command returns the vertical scale factor per digitizing level in units specified by WFMOutpre:YUNit for the waveform specified by the [DATA:SOURce](#) command. For those formats in which [WFMOutpre:BYT\\_Nr](#) is important (all non-floating point formats), WFMOutpre:YMult? must take the location of the binary point implied by BYT\_NR into consideration. An error is generated if the waveform specified by [DATA:SOURce](#) is not turned on.

**Group** Waveform Transfer

**Syntax** WFMOutpre:YMult?

**Related Commands** [DATA:SOURce](#)

**Examples** WFMOUTPRE:YMULT? might return :WFMOUTPRE:YMULT 4.0000E-3 indicating that the vertical scale for the corresponding waveform is 100 mV/div.

## WFMOutpre:YOff? (Query Only)

This query-only command returns the vertical offset in digitized levels for the waveform specified by the [DATA:SOURce](#) command. For those formats in which BYT\_NR is important (all non-floating point formats), this command must take the location of the binary point implied by [WFMOutpre:BYT\\_Nr](#) into consideration. An error is generated if the waveform specified by [DATA:SOURce](#) is not turned on.

**Group** Waveform Transfer

**Syntax** WFMOutpre:YOff?

**Related Commands** [DATA:SOURce](#), [WFMOutpre:BYT\\_Nr](#)

**Examples** WFMOUTPRE:YOFF? might return :WFMOUTPRE:YOFF -50.0000E+0 indicating that the position indicator for the waveform was 50 digitizing levels (2 divisions) below center screen.



## WFMOutpre:YUNit? (Query Only)

This query-only command returns the vertical units for the waveform specified by the [DATA:SOURce](#) command. An error is generated if the waveform specified by [DATA:SOURce](#) is not turned on.

**Group** Waveform Transfer

**Syntax** WFMOutpre:YUNit?

**Related Commands** [DATA:SOURce](#)

**Examples** WFMOUTPRE:YUNIT? might return :WFMOUTPRE:YUNIT "dB" indicating that the vertical units for the waveform are measured in decibels.

## WFMOutpre:YZero? (Query Only)

This query-only command returns the vertical offset in units specified by [WFMOutpre:YUNit?](#) for the waveform specified by the [DATA:SOURce](#) command. An error is generated if the waveform specified by [DATA:SOURce](#) is not turned on.

**Group** Waveform Transfer

**Syntax** WFMOutpre:YZero?

**Related Commands** [DATA:SOURce](#), [WFMOutpre:YUNit?](#)

**Examples** WFMOUTPRE:YZERO? might return :WFMOUTPRE:YZERO -100.0000E-3 indicating that vertical offset is set to -100 mV.

## ZOOM? (Query Only)

This command resets the zoom transforms to default values for all traces or live traces. The ZOOM query returns the current vertical and horizontal positioning and scaling of the display.

**Group** Zoom

**Syntax**    ZOOM?

**Examples**    ZOOM?? might return :ZOOM:MODE 0;GRATICULE:SIZE 80;SPLIT EIGHTYTWENTY;ZOOM:SCROLL:DIRECTION STOP; LOCK 0;SPEED 1;ZOOM:ZOOM1:STATE 1; SCROLLLOCK 1;CH1:HORIZONTAL:POSITION 50.0000; SCALE 5:ZOOM:ZOOM1:CH1:VERTICAL:POSITION 0.0000; SCALE 1.0000:ZOOM:ZOOM1:CH2:HORIZONTAL:POSITION 50.0000;SCALE 5:ZOOM:ZOOM1:CH2:VERTICAL:POSITION 0.0000; SCALE 1.0000:ZOOM:ZOOM1:CH3:HORIZONTAL:POSITION 50.0000; SCALE 5:ZOOM:ZOOM1:CH3:VERTICAL:POSITION 0.0000; SCALE 1.0000:ZOOM:ZOOM1:CH4:HORIZONTAL:POSITION 50.0000; SCALE 5:ZOOM:ZOOM1:CH4:VERTICAL:POSITION 0.0000;SCALE 1.0000:ZOOM:ZOOM1:MATH1:HORIZONTAL:POSITION 50.0000; SCALE 5:ZOOM:ZOOM1:MATH1:VERTICAL:POSITION 0.0000;SCALE 1.0000:ZOOM:ZOOM1:MATH2:HORIZONTAL:POSITION 50.0000; SCALE 5:ZOOM:ZOOM1:MATH2:VERTICAL:POSITION 0.0000;...

## ZOOM:GRATICULE:SIZE? (Query Only)

This query returns the size, in percent, of the Zoom (lower) graticule. The query always returns 80.

**Group**    Zoom

**Syntax**    ZOOM:GRATICULE:SIZE??

**Examples**    ZOOM:GRATICULE:SIZE?? always returns :ZOOM:GRATICULE:SIZE 80 indicating that the zoom graticule size is set to 80%.

## ZOOM:GRATICULE:SPLIT? (Query Only)

This query returns the sizes of the acquisition and zoom windows when Zoom is selected. The query always returns EIGHTYTWENTY.

**Group**    Zoom

**Syntax**    ZOOM:GRATICULE:SPLIT??

**Related Commands**    [ZOOM{:MODE|:STATE}](#)

**Examples** ZOOM:GRATICULE:SPLIT?? always returns :ZOOM:GRATICULE:SPLIT EIGHTYTWENTY, indicating that the display area is divided 80%/20% between the zoomed graticule and the acquisition graticule.

## ZOOM{:MODE|:STATE}

This command turns Zoom mode on or off. The Zoom query returns the current state of Zoom mode. This command is equivalent to pressing the **ZOOM** button located on the front panel.

**Group** Zoom

**Syntax** ZOOM{:MODE|:STATE} {ON|OFF|<NR1>}  
ZOOM{:MODE|:STATE}

**Arguments** ON turns on Zoom mode.  
OFF turns off Zoom mode.  
<NR1> = 0 turns off Zoom mode; any other value turns on Zoom mode.

**Examples** ZOOM:MODE OFF turns off Zoom mode.  
ZOOM:MODE? might return :ZOOM:MODE 1 indicating that Zoom mode is currently turned on.

## ZOOM:ZOOM<x>? (Query Only)

This command resets the zoom transforms to default values for all traces of the specified zoom, where x is an integer from 1 to 4 representing the desired zoom window. The ZOOM? query returns the current vertical and horizontal positioning and scaling of the display.

**Group** Zoom

**Syntax** ZOOM:ZOOM<x>?

**Examples** ZOOM:ZOOM1? might return :ZOOM:ZOOM1:STATE 1; SCROLLLOCK 1; CH1:HORIZONTAL POSITION 50.0000; SCALE 5;  
:ZOOM:ZOOM1:CH1:VERTICAL:POSITION 0.0000; SCALE 1.0000; :ZOOM:ZOOM1:CH2:HORIZONTAL:POSITION 50.0000;

SCALE 5; :ZOOM:ZOOM1:CH2:VERTICAL:POSITION 0.000; SCALE 1.0000; :ZOOM:ZOOM1:CH3:HORIZONTAL:POSITION50.000; SCALE 5; :ZOOM:ZOOM1:CH3:VERTICAL... indicating the Zoom1 transforms for all traces.

## ZOOM:ZOOM<x>:FACTOR? (Query Only)

This query-only command returns the zoom factor of a particular zoom box.

|                |   |
|----------------|---|
| <b>Group</b>   | Zoom                                    |
| <b>Syntax</b>  | ZOOM:ZOOM<x>:FACTOR?                    |
| <b>Returns</b> | <NR1> is the zoom factor of a zoom box. |

## ZOOM:ZOOM<x>:HORIZONTAL:POSITION

This command sets or returns the horizontal position for the specified zoom, where x is an integer from 1 to 4 representing the desired zoom window.

|                  |  |
|------------------|--|
| <b>Group</b>     | Zoom   |
| <b>Syntax</b>    | ZOOM:ZOOM<x>:HORIZONTAL:POSITION <NR3><br>ZOOM:ZOOM<x>:HORIZONTAL:POSITION?  |
| <b>Arguments</b> | <NR3> is a value from 0 to 100.00 and is the percent of the waveform that is to the left of screen center, when the zoom factor is 1× or greater.  |
| <b>Examples</b>  | ZOOM:ZOOM1:HORIZONTAL:POSITION 50 sets the Zoom1 reference pointer at 50% of acquired waveform.<br><br>ZOOM:ZOOM1:HORIZONTAL:POSITION? might return :ZOOM1:HORIZONTAL:POSITION 50.0000 indicating that the Zoom1 reference pointer is currently set at 50% of acquired waveform. |

## ZOOM:ZOOM<x>:HORIZONTAL:SCALE

This command sets or returns the zoom horizontal scale factor for the specified zoom, where x is an integer from 1 to 4 representing the desired zoom window.

|                  |   |
|------------------|---|
| <b>Group</b>     | Zoom  |
| <b>Syntax</b>    | ZOOM:ZOOM<x>:HORizontal:SCALE <NR3><br>ZOOM:ZOOM<x>:HORizontal:SCALE?   |
| <b>Arguments</b> | <NR3> is the amount of expansion in the horizontal direction in 1-2 -5 increments.  |
| <b>Examples</b>  | <p>ZOOM:ZOOM1:HORIZONTAL:SCALE 5 sets the horizontal scale factor of Channel 1 to 5.</p> <p>ZOOM:ZOOM2:HORIZONTAL:SCALE? might return<br/> :ZOOM2:HORIZONTAL:SCALE 1, indicating that the horizontal scale factor is 1.</p> |

## ZOOM:ZOOM<x>:POSition

This command sets the horizontal position of a particular zoom box, in terms of 0 to 100.0% of upper window

|                  |  |
|------------------|--|
| <b>Group</b>     | Zoom   |
| <b>Syntax</b>    | ZOOM:ZOOM<x>:POSition <NR3><br>ZOOM:ZOOM<x>:POSition?              |
| <b>Arguments</b> | <NR3> is the horizontal position as a percent of the upper window. |

## ZOOM:ZOOM<x>:SCALE

This command sets or returns the horizontal scale of the particular zoom box.

|                  |   |
|------------------|---|
| <b>Group</b>     | Zoom  |
| <b>Syntax</b>    | ZOOM:ZOOM<x>:SCALE <NR3><br>ZOOM:ZOOM<x>:SCALE? |
| <b>Arguments</b> | <NR3> is the horizontal scale of the zoom box.  |

## ZOOM:ZOOM<x>:STATE

This command sets or returns the specified zoom on or off, where x is an integer from 1 to 4 representing the desired zoom window.

Default is Zoom1 on, Zoom2 - 4 off.

**Group** Zoom

**Syntax** ZOOM:ZOOM<x>:STATE {ON|OFF|<NR1>}  
ZOOM:ZOOM<x>:STATE?

**Arguments** ON turns Zoom 1-4 on.  
OFF turns Zoom 1-4 off.  
<NR1> = 0 disables the specified zoom; any other value enables the specified zoom.

**Examples** ZOOM:ZOOM<x>:STATE ON turns Zoom1 on.  
ZOOM:ZOOM<x>:STATE? might return :ZOOM:ZOOM2:STATE 1 indicating that Zoom2 is on.

# Status and Events

The instrument provides a status and event reporting system for the GPIB interfaces. This system informs you of certain significant events that occur within the instrument.

The instrument status handling system consists of five 8-bit registers and two queues for each interface. The remaining Status subtopics describe these registers and components. They also explain how the event handling system operates.

## Registers

### Overview

The registers in the event handling system fall into two functional groups:

- Status Registers contain information about the status of the instrument. They include the Standard Event Status Register (SESR)
- Enable Registers determine whether selected types of events are reported to the Status Registers and the Event Queue. They include the Device Event Status Enable Register (DESER), the Event Status Enable Register (ESER), and the Service Request Enable Register (SRER).

### Status Registers

The Standard Event Status Register (SESR) and the Status Byte Register (SBR) record certain types of events that may occur while the instrument is in use. IEEE Std 488.2-1987 defines these registers.

Each bit in a Status Register records a particular type of event, such as an execution error or message available. When an event of a given type occurs, the instrument sets the bit that represents that type of event to a value of one. (You can disable bits so that they ignore events and remain at zero. See Enable Registers). Reading the status registers tells you what types of events have occurred.

**The Standard Event Status Register (SESR).** The SESR records eight types of events that can occur within the instrument. Use the \*ESR? query to read the SESR register. Reading the register clears the bits of the register so that the register can accumulate information about new events.

---

**NOTE.** TekVISA applications use SESR bit 6 to respond to any of several events, including some front panel actions.

---

|     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
| PON | URQ | CME | EXE | DDE | QYE | RQC | OPC |

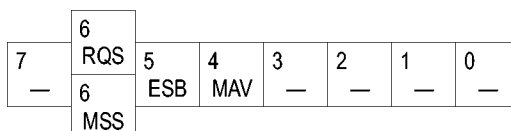
**Figure 3-1: The Standard Event Status Register (SESR)**

**Table 3-1: SESR Bit Functions**

| Bit     | Function |  |
|---------|----------|--|
| 7 (MSB) | PON      | Power On. Shows that the oscilloscope was powered on. On completion, the diagnostic tests also set this bit.                                     |
| 6       | URQ      | User Request. Indicates that an application event has occurred. *See note.   |
| 5       | CME      | Command Error. Shows that an error occurred while the oscilloscope was parsing a command or query.   |
| 4       | EXE      | Execution Error. Shows that an error executing a command or query.   |
| 3       | DDE      | Device Error. Shows that a device error occurred.  |
| 2       | QYE      | Query Error. Either an attempt was made to read the Output Queue when no data was present or pending, or that data in the Output Queue was lost. |
| 1       | RQC      | Request Control. This is not used.   |
| 0 (LSB) | OPC      | Operation Complete. Shows that the operation is complete. This bit is set when all pending operations complete following an *OPC command.        |

**The Status Byte Register (SBR).** Records whether output is available in the Output Queue, whether the instrument requests service, and whether the SESR has recorded any events.

Use a Serial Poll or the \*STB? query to read the contents of the SBR. The bits in the SBR are set and cleared depending on the contents of the SESR, the Event Status Enable Register (ESER), and the Output Queue. When you use a Serial Poll to obtain the SBR, bit 6 is the RQS bit. When you use the \*STB? query to obtain the SBR, bit 6 is the MSS bit. Reading the SBR does not clear the bits.

**Figure 3-2: The Status Byte Register (SBR)****Table 3-2: SBR Bit Functions**

| Bit     | Function |  |
|---------|----------|--|
| 7 (MSB) | —        | Not used.  |
| 6       | RQS      | Request Service. Obtained from a serial poll. Shows that the oscilloscope requests service from the GPIB controller. |
| 6       | MSS      | Master Status Summary. Obtained from *STB? query. Summarizes the ESB and MAV bits in the SBR.                        |



**Table 3-2: SBR Bit Functions, (cont.)**

| Bit | Function   |
|-----|--|
| 5   | ESB      Event Status Bit. Shows that status is enabled and present in the SESR. |
| 4   | MAV      Message Available. Shows that output is available in the Output Queue.  |
| 3   | ———— Not used.   |
| 2   | ———— Not used.   |
| 1–0 | ———— Not used.   |

## Enable Registers

DESER, ESER, and SRER allow you to select which events are reported to the Status Registers and the Event Queue. Each Enable Register acts as a filter to a Status Register (the DESER also acts as a filter to the Event Queue) and can prevent information from being recorded in the register or queue.

Each bit in an Enable Register corresponds to a bit in the Status Register it controls. In order for an event to be reported to a bit in the Status Register, the corresponding bit in the Enable Register must be set to one. If the bit in the Enable Register is set to zero, the event is not recorded.

Various commands set the bits in the Enable Registers. The Enable Registers and the commands used to set them are described below.

**The Device Event Status Enable Register (DESER).** This register controls which types of events are reported to the SESR and the Event Queue. The bits in the DESER correspond to those in the SESR.

Use the DESE command to enable and disable the bits in the DESER. Use the DESE? query to read the DESER.

|     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
| PON | URQ | CME | EXE | DDE | QYE | RQC | OPC |

**Figure 3-3: The Device Event Status Enable Register (DESER)**

**The Event Status Enable Register (ESER).** This register controls which types of events are summarized by the Event Status Bit (ESB) in the SBR. Use the \*ESE command to set the bits in the ESER. Use the \*ESE? query to read it.

|     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
| PON | URQ | CME | EXE | DDE | QYE | RQC | OPC |

**Figure 3-4: The Event Status Enable Register (ESER)**

**The Service Request Enable Register (SRER).** This register controls which bits in the SBR generate a Service Request and are summarized by the Master Status Summary (MSS) bit.

Use the \*SRE command to set the SRER. Use the \*SRE? query to read the register. The RQS bit remains set to one until either the Status Byte Register is read with a Serial Poll or the MSS bit changes back to a zero.

|   |   |     |     |   |   |   |   |
|---|---|-----|-----|---|---|---|---|
| 7 | 6 | 5   | 4   | 3 | 2 | 1 | 0 |
| — | — | ESB | MAV | — | — | — | — |

**Figure 3-5: The Service Request Enable Register (SRER)**

### \*PSC Command

The \*PSC command controls the Enable Registers contents at power-on. Sending \*PSC 1 sets the Enable Registers at power on as follows:

- DESER 255 (equivalent to a DESe 255 command)
- ESER 0 (equivalent to an \*ESE 0 command)
- SRER 0 (equivalent to an \*SRE 0 command)

Sending \*PSC 0 lets the Enable Registers maintain their values in nonvolatile memory through a power cycle.

Note: To enable the PON (Power On) event to generate a Service Request, send \*PSC 0, use the DESe and \*ESE commands to enable PON in the DESER and ESER, and use the \*SRE command to enable bit 5 in the SRER. Subsequent power-on cycles will generate a Service Request.

## Queues

The \*PSC command controls the Enable Registers contents at power-on. Sending \*PSC 1 sets the Enable Registers at power on as follows:

### Output Queue

The instrument stores query responses in the Output Queue and empties this queue each time it receives a new command or query message after an <EOM>. The controller must read a query response before it sends the next command (or query) or it will lose responses to earlier queries.



**CAUTION.** When a controller sends a query, an <EOM>, and a second query, the instrument normally clears the first response and outputs the second while reporting a Query Error (QYE bit in the ESER) to indicate the lost response. A fast controller, however, may receive a part or all of the first response as well. To avoid this situation, the controller should always read the response immediately after sending any terminated query message or send a DCL (Device Clear) before sending the second query.

**Event Queue**

The Event Queue stores detailed information on up to 33 events. If more than 32 events stack up in the Event Queue, the 32nd event is replaced by event code 350, "Queue Overflow."

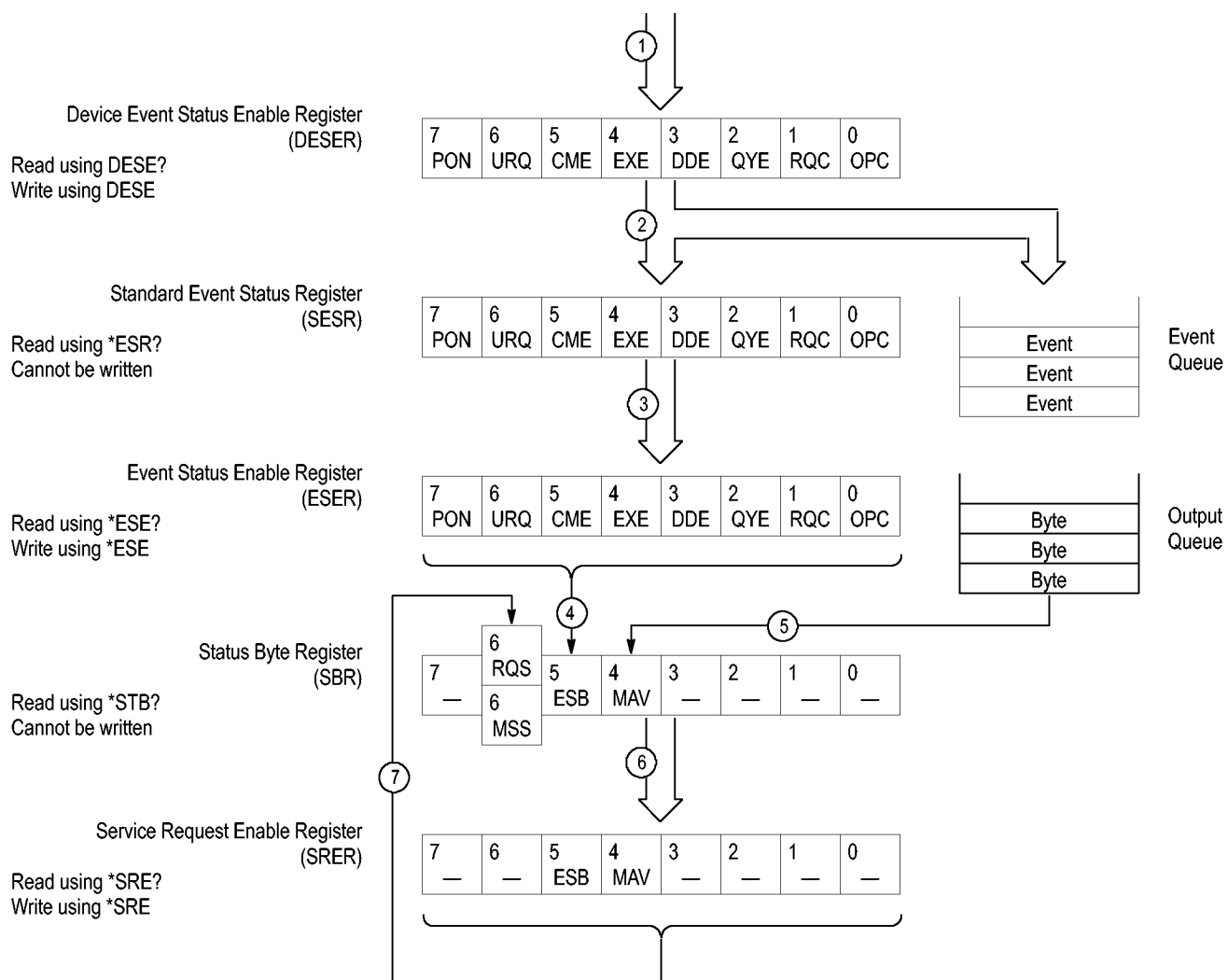
Read the Event Queue with the EVENT? query (which returns only the event number), with the EVMSG? query (which returns the event number and a text description of the event), or with the ALLEV? query (which returns all the event numbers along with a description of the event). Reading an event removes it from the queue.

Before reading an event from the Event Queue, you must use the \*ESR? query to read the summary of the event from the SESR. This makes the events summarized by the \*ESR? read available to the EVENT? and EVMSG? queries, and empties the SESR.

Reading the SESR erases any events that were summarized by previous \*ESR? reads but not read from the Event Queue. Events that follow an \*ESR? read are put in the Event Queue but are not available until \*ESR? is used again.

## Event Handling Sequence

The figure below shows how to use the status and event handling system. In the explanation that follows, numbers in parentheses refer to numbers in the figure.



**Figure 3-6: Status and Event Handling Process**

When an event occurs, a signal is sent to the DESER (1). If that type of event is enabled in the DESER (that is, if the bit for that event type is set to 1), the appropriate bit in the SESR is set to one, and the event is recorded in the Event Queue (2). If the corresponding bit in the ESER is also enabled (3), then the ESB bit in the SBR is set to one (4).

When output is sent to the Output Queue, the MAV bit in the SBR is set to one (5).

When a bit in the SBR is set to one and the corresponding bit in the SRER is enabled (6), the MSS bit in the SBR is set to one and a service request is generated (7).

## Synchronization Methods

### Overview

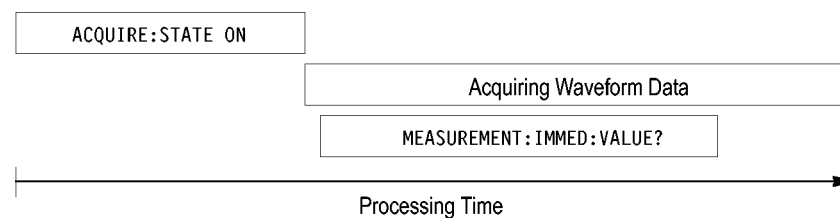
Although most GPIB commands are completed almost immediately after being received by the instrument, some commands start a process that requires more time. For example, once a single sequence acquisition command is executed, depending upon the applied signals and trigger settings, it may be a few seconds before the acquisition is complete. Rather than remain idle while the operation is in process, the instrument will continue processing other commands. This means that some operations will not be completed in the order that they were sent.

Sometimes the result of an operation depends on the result of an earlier operation. A first operation must complete before the next one gets processed. The instrument status and event reporting system provides ways to do this.

For example, a typical application might involve acquiring a single-sequence waveform and then taking a measurement on the acquired waveform. You could use the following command sequence to do this:

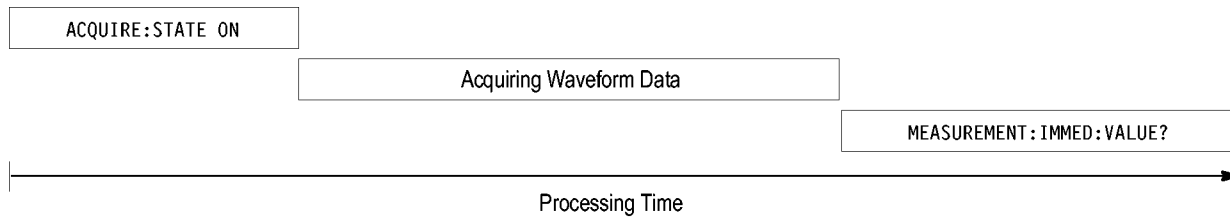
```
/** Set up conditional acquisition **/
ACQUIRE:STATE OFF
SELECT:CH1 ON
HORIZONTAL:RECORDLENGTH 1000
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/** Acquire waveform data **/
ACQUIRE:STATE ON
/** Set up the measurement parameters **/
MEASUREMENT:IMMED:TYPE AMPLITUDE
MEASUREMENT:IMMED:SOURCE CH1
/** Take amplitude measurement **/
MEASUREMENT:MEAS1:VALUE
```

The acquisition of the waveform requires extended processing time. It may not finish before the instrument takes an amplitude measurement (see the following figure). This can result in an incorrect amplitude value.



**Figure 3-7: Command Processing Without Using Synchronization**

To ensure the instrument completes waveform acquisition before taking the measurement on the acquired data, you can synchronize the program.



**Figure 3-8: Processing Sequence With Synchronization**

You can use four commands to synchronize the operation of the instrument with your application program: `*WAI`, `BUSY`, `*OPC`, and `*OPC`

### Using the `*WAI` Command

The `*WAI` command forces completion of previous commands that generate an OPC message. No commands after the `*WAI` are processed before the OPC message(s) are generated

The same command sequence using the `*WAI` command for synchronization looks like this:

```

/* Set up conditional acquisition */
ACQUIRE:STATE OFF
SELECT:CH1 ON
HORIZONTAL:RECORDLENGTH 1000
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/* Acquire waveform data */
ACQUIRE:STATE ON
/* Set up the measurement parameters */
MEASUREMENT:IMMED:TYPE AMPLITUDE
MEASUREMENT:IMMED:SOURCE CH1
/* wait until the acquisition is complete before taking
the measurement*/
*/
*WAI
/* Take amplitude measurement */
MEASUREMENT:IMMED:VALUE
    
```

The controller can continue to write commands to the input buffer of the instrument, but the commands will not be processed by the instrument until all OPC operations in process are complete. If the input buffer becomes full, the controller will be unable to write more commands to the buffer. This can cause a time-out.

### Using the `BUSY` Query

The `BUSY?` query allows you to find out whether the instrument is busy processing a command that has an extended processing time such as single-sequence acquisition.

The same command sequence, using the BUSY? query for synchronization, looks like this:

```
/* Set up conditional acquisition */
ACQUIRE:STATE OFF
SELECT:CH1 ON
HORIZONTAL:RECORDLENGTH 1000
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/* Acquire waveform data */
ACQUIRE:STATE ON
/* Set up the measurement parameters */
MEASUREMENT:IMMED:TYPE AMPLITUDE
MEASUREMENT:IMMED:SOURCE CH1
/* wait until the acquisition is complete before taking
the measurement */
while BUSY keep looping
/* Take amplitude measurement */
MEASUREMENT:IMMED:VALUE
```

This sequence lets you create your own wait loop rather than using the \*WAI command. The BUSY? query helps you avoid time-outs caused by writing too many commands to the input buffer. The controller is still tied up though, and the repeated BUSY? query will result in more bus traffic.

## Using the \*OPC Command

If the corresponding status registers are enabled, the \*OPC command sets the OPC bit in the Standard Event Status Register (SESR) when an operation is complete. You achieve synchronization by using this command with either a serial poll or service request handler.

**Serial Poll Method:** Enable the OPC bit in the Device Event Status Enable Register (DESER) and the Event Status Enable Register (ESER) using the DESE and \*ESE commands.

When the operation is complete, the OPC bit in the Standard Event Status Register (SESR) will be enabled and the Event Status Bit (ESB) in the Status Byte Register will be enabled.

The same command sequence using the \*OPC command for synchronization with serial polling looks like this:

```
/* Set up conditional acquisition */
ACQUIRE:STATE OFF
SELECT:CH1 ON
HORIZONTAL:RECORDLENGTH 1000
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/* Enable the status registers */
DESE 1
```

```
*ESE 1
*SRE 0
/* Acquire waveform data */
ACQUIRE:STATE ON
/* Set up the measurement parameters */
MEASUREMENT:IMMED:TYPE AMPLITUDE
MEASUREMENT:IMMED:SOURCE CH1
/* wait until the acquisition is complete before taking the
measurement.*/
*OPC
while serial poll = 0, keep looping
/* Take amplitude measurement */
MEASUREMENT:IMMED:VALUE
```

This technique requires less bus traffic than did looping on BUSY.

**Service Request Method:** Enable the OPC bit in the Device Event Status Enable Register (DESER) and the Event Status Enable Register (ESER) using the DESE and \*ESE commands.

You can also enable service requests by setting the ESB bit in the Service Request Enable Register (SRER) using the \*SRE command. When the operation is complete, a Service Request will be generated.

The same command sequence using the \*OPC command for synchronization looks like this

```
/* Set up conditional acquisition */
ACQUIRE:STATE OFF
SELECT:CH1 ON
HORIZONTAL:RECORDLENGTH 1000
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/* Enable the status registers */
DESE 1
*ESE 1
*SRE 32
/* Acquire waveform data */
ACQUIRE:STATE ON
/* Set up the measurement parameters */
MEASUREMENT:IMMED:TYPE AMPLITUDE
MEASUREMENT:IMMED:SOURCE CH1
/* wait until the acquisition is complete before taking the
measurement*/
*OPC
```

The program can now do different tasks such as talk to other devices. The SRQ, when it comes, interrupts those tasks and returns control to this task.

```
/* Take amplitude measurement */
MEASUREMENT:IMMED:VALUE
```



This technique is more efficient but requires more sophisticated programming.

### Using the \*OPC? Query

The \*OPC? query places a 1 in the Output Queue once an operation that generates an OPC message is complete. A time out could occur if you try to read the output queue before there is any data in it.

The same command sequence using the \*OPC? query for synchronization looks like this:

```
/* Set up conditional acquisition */
ACQUIRE:STATE OFF
SELECT:CH1 ON
HORIZONTAL:RECORDLENGTH 1000
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/* Acquire waveform data */
ACQUIRE:STATE ON
/* Set up the measurement parameters */
MEASUREMENT:IMMED:TYPE AMPLITUDE
MEASUREMENT:IMMED:SOURCE CH1
/* wait until the acquisition is complete before taking the
measurement*/
*OPC
```

Wait for read from Output Queue.

```
/* Take amplitude measurement */
MEASUREMENT:IMMED:VALUE
```

This is the simplest approach. It requires no status handling or loops. However, you must set the controller time-out for longer than the acquisition operation.

## Messages

### Overview

The information contained in the topic tabs above covers all the programming interface messages the instrument generates in response to commands and queries.

For most messages, a secondary message from the instrument gives more detail about the cause of the error or the meaning of the message. This message is part of the message string and is separated from the main message by a semicolon.

Each message is the result of an event. Each type of event sets a specific bit in the SESR and is controlled by the equivalent bit in the DESER. Thus, each message is associated with a specific SESR bit. In the message tables, the associated SESR bit is specified in the table title, with exceptions noted with the error message text.

### No Event

The following table shows the messages when the system has no events or status to report. These have no associated SESR bit.

**Table 3-3: No Event Messages**

| Code | Message                                       |
|------|---|
| 0    | No events to report; queue empty              |
| 1    | No events to report; new events pending *ESR? |

**Command Error**

The following table shows the command error messages generated by improper syntax. Check that the command is properly formed and that it follows the rules in the section on command Syntax.

**Table 3-4: Command Error Messages (CME Bit 5)**

| Code | Message                     |
|------|-----------------------------|
| 100  | Command error               |
| 101  | Invalid character           |
| 102  | Syntax error                |
| 103  | Invalid separator           |
| 104  | Data type error             |
| 105  | GET not allowed             |
| 108  | Parameter not allowed       |
| 109  | Missing parameter           |
| 110  | Command header error        |
| 111  | Header separator error      |
| 112  | Program mnemonic too long   |
| 113  | Undefined header            |
| 114  | Header suffix out of range  |
| 120  | Numeric data error          |
| 121  | Invalid character in number |
| 123  | Exponent too large          |
| 124  | Too many digits             |
| 128  | Numeric data not allowed    |
| 130  | Suffix error                |
| 131  | Invalid suffix              |
| 134  | Suffix too long             |
| 138  | Suffix not allowed          |
| 140  | Character data error        |
| 141  | Invalid character data      |
| 144  | Character data too long     |
| 148  | Character data not allowed  |
| 150  | String data error           |
| 151  | Invalid string data         |

**Table 3-4: Command Error Messages (CME Bit 5), (cont.)**

| <b>Code</b> | <b>Message</b>              |
|-------------|-----------------------------|
| 158         | String data not allowed     |
| 160         | Block data error            |
| 161         | Invalid block data          |
| 168         | Block data not allowed      |
| 170         | Command expression error    |
| 171         | Invalid expression          |
| 178         | Expression data not allowed |

**Execution Error** The following table lists the execution errors that are detected during execution of a command.

**Table 3-5: Execution Error Messages (EXE Bit 4)**

| <b>Code</b> | <b>Message</b>              |
|-------------|-----------------------------|
| 200         | Execution error             |
| 201         | Invalid while in local      |
| 202         | Settings lost due to RTL    |
| 210         | Trigger error               |
| 211         | Trigger ignored             |
| 212         | Arm ignored                 |
| 219         | Trigger delay not available |
| 220         | Parameter error             |
| 221         | Settings conflict           |
| 222         | Data out of range           |
| 223         | Too much data               |
| 224         | Illegal parameter value     |
| 225         | Out of memory               |
| 230         | Data corrupt or stale       |
| 240         | Hardware error              |
| 241         | Hardware missing            |
| 244         | Invalid parameter selected  |
| 250         | Mass storage error          |
| 251         | Missing mass storage        |
| 252         | Missing media               |
| 253         | Corrupt media               |
| 254         | Media full                  |
| 255         | Directory full              |

**Table 3-5: Execution Error Messages (EXE Bit 4), (cont.)**

| <b>Code</b> | <b>Message</b>  |
|-------------|---|
| 256         | File name not found                                     |
| 257         | File name error   |
| 258         | Media protected   |
| 260         | Execution expression error                              |
| 261         | Math error in expression                                |
| 2200        | Measurement error, Measurement system error             |
| 2201        | Measurement error, No period found                      |
| 2203        | Measurement error, No period, second waveform           |
| 2204        | Measurement error, Low signal amplitude                 |
| 2205        | Measurement error, Low amplitude, second waveform       |
| 2206        | Measurement error, Invalid gate                         |
| 2207        | Measurement error, Measurement overflow                 |
| 2208        | Measurement error, Waveform does not cross Mid Ref      |
| 2209        | Measurement error, No second Mid Ref crossing           |
| 2210        | Measurement error, No Mid Ref crossing, second waveform |
| 2211        | Measurement error, No backwards Mid Ref Crossing        |
| 2212        | Measurement error, No negative crossing                 |
| 2213        | Measurement error, No positive crossing                 |
| 2214        | Measurement error, No crossing                          |
| 2215        | Measurement error, No crossing, second waveform         |
| 2216        | Measurement error, No crossing, target waveform         |
| 2217        | Measurement error, Constant waveform                    |
| 2218        | Measurement error, Unused                               |
| 2219        | Measurement error, No valid edge – No arm sample        |
| 2220        | Measurement error, No valid edge – No arm cross         |
| 2221        | Measurement error, No valid edge – No trigger cross     |
| 2222        | Measurement error, No valid edge – No second cross      |
| 2223        | Measurement error, Waveform mismatch                    |
| 2225        | Measurement error, No waveform to measure               |
| 2226        | Measurement error, Null Waveform                        |
| 2227        | Measurement error, Positive and Negative Clipping       |
| 2228        | Measurement error, Positive Clipping                    |
| 2229        | Measurement error, Negative Clipping                    |
| 2230        | Measurement error, High Ref < Low Ref                   |
| 2231        | Measurement error, no statistics available              |
| 2241        | Waveform request is invalid                             |

**Table 3-5: Execution Error Messages (EXE Bit 4), (cont.)**

| <b>Code</b> | <b>Message</b>  |
|-------------|---|
| 2243        | This measurement cannot be performed on this type of waveform                 |
| 2244        | Source waveform is not active   |
| 2248        | This ref cannot be activated  |
| 2249        | Reference deletion error, Waveform in use for math                            |
| 2250        | Reference error, Waveform reference file is invalid                           |
| 2251        | Reference error, Waveform preamble does not match actual                      |
| 2252        | Reference error, Source waveform is not valid                                 |
| 2253        | Reference error, Too many points received                                     |
| 2254        | Reference error, Too few points received                                      |
| 2259        | File too big  |
| 2261        | Calibration error, wait for the warmup interval to expire before invoking SPC |
| 2400        | Not enough memory available   |
| 2401        | This channel cannot be activated  |
| 2402        | Math/Meas/Histo have circular definition                                      |
| 2410        | Empty math string   |
| 2411        | Syntax error in math string   |
| 2412        | Semantic error in math string   |
| 2413        | Math expression is too complex  |
| 2420        | Histogram cannot be performed on this type of waveform                        |

**Device Error**

The following table lists the device errors that can occur during instrument operation. These errors may indicate that the instrument needs repair.

**Table 3-6: Device Error Messages (DDE Bit 3)**

| <b>Code</b> | <b>Message</b>                        |
|-------------|---------------------------------------|
| 300         | Device-specific error                 |
| 310         | System error                          |
| 311         | Memory error                          |
| 312         | PUD memory lost                       |
| 313         | Calibration memory lost               |
| 314         | Save/recall memory los                |
| 315         | Configuration memory lost             |
| 350         | Queue overflow (does not set DDE bit) |

**System Event**

The following table lists the system event messages. These messages are generated whenever certain system conditions occur.

**Table 3-7: System Event Messages**

| Code | Message  |
|------|--|
| 400  | Query event  |
| 401  | Power on (PON bit 7 set)                                     |
| 402  | Operation complete (OPC bit 0 set)                           |
| 403  | User request (URQ bit 6 set)                                 |
| 404  | Power fail (DDE bit 3 set)                                   |
| 405  | Request control  |
| 410  | Query INTERRUPTED (QYE bit 2 set)                            |
| 420  | Query UNTERMINATED (QYE bit 2 set)                           |
| 430  | Query DEADLOCKED (QYE bit 2 set)                             |
| 440  | Query UNTERMINATED after indefinite response (QYE bit 2 set) |

**Execution Warning**

The following table lists warning messages that do not interrupt the flow of command execution. These notify you that you may get unexpected results.

**Table 3-8: Execution Warning Messages (EXE Bit 4)**

| Code | Message   |
|------|---|
| 540  | Measurement warning                             |
| 541  | Measurement warning, Low signal amplitude       |
| 542  | Measurement warning, Unstable histogram         |
| 543  | Measurement warning, Low resolution             |
| 544  | Measurement warning, Uncertain edge             |
| 545  | Measurement warning, Invalid min max            |
| 546  | Measurement warning, Need 3 edges               |
| 547  | Measurement warning, Clipping positive/negative |
| 548  | Measurement warning, Clipping positive          |
| 549  | Measurement warning, Clipping negative          |
| 551  | FASTAcq mode is active – deactivate to use math |

**Internal Warning**

The following table shows internal errors that indicate an internal fault in the instrument.

**Table 3-9: Internal Warning Messages**

| Code | Message                        |
|------|--------------------------------|
| 600  | Internal warning               |
| 630  | Internal warning, 50Ω overload |

---

# Programming Examples

## Programming Examples

### Overview

Three example programs, which demonstrate methods that you can use to control the instrument through the General Purpose Interface Bus (GPIB), are included on your *DPO4000 Series Product Software* CD. These example programs are installed as part of the *GPIB Programmer* installation, which includes the DPO4000 Series Oscilloscope Programmer Online Guide.

To install the "GPIB Programmer", perform the following procedure:

1. Insert the *DPO4000 Series Product Software* CD in your CD drive.
2. Open the folder named *GPIB Programmer*.
3. Double-click *Setup.exe*.
4. Follow the Install Wizard directions.

An Examples directory will be created with the following path name: *C:\Program Files\TekScope\Programmer\Examples*

Within the *Examples* directory are two subdirectories, *Source* and *Programs*.

Source contains the source files, written in Microsoft Visual C++ , Version 6.0, which is required to build executable files for each of the following example programs:

- Meas.c – This example program demonstrates how to perform a periodic amplitude measurement on CH 1 of your instrument.
- GetWfm.c – This example program demonstrates how to acquire and output the CH 1 waveform in a 500-point RIBANARY format. The program then queries the instrument to get the waveform preamble information, formats the binary waveform data as ASCII values, and, finally, writes out a report of the waveform preamble and ASCII data points to a file named WFM\_DATA.PRN.
- Tl.c – This example program demonstrates how to use a general talker/listener program to allow the user to send commands and queries to the instrument and display the responses.

A *README* file in the *Source* directory has the latest documentation.

Programs contains compiled, executable files for each of the example programs in the Source directory (see above).

The programs run on a PC-compatible system equipped with a National Instruments GPIB board and associated drivers.

## Compiling GPIB Programs

The example GPIB programs make the following assumptions:

- The GPIB controller board is "GPIB0" (board 0). If you have more than one GPIB controller in your workstation, and you want to use a GPIB controller other than board 0, you must edit the source files accordingly.
- The instrument is connected to the GPIB controller (see above) and is set to address 1 (DEV1). If you want to use another address, then you must edit the source files accordingly.

Each program requires adding the following source files to the associated VC++ project (which you will create).

- *GPIBERR.C*
- *GPIBREAD.C*
- *GPIBWAIT.C*
- *GPIBWRT.C*

In addition to the above source files, you will also need to add the following files provided by National Instruments:

- *decl-32.h*
- *gpib-32.obj*

## Visual C++

To make an executable for any of the example programs in the Source directory, perform the following steps:

1. Install Microsoft Visual C++, Version 6.0.
2. Install the National Instruments GPIB board and drivers.
3. Copy the following source files from your National Instruments GPIB drivers directory to this directory ("Examples\Source"):
  - *decl-32.h*
  - *gpib-32.obj*
1. Create a new "Win32 Console Application" project in Visual C++ located in this directory ("Examples\Source"). For example, if you want to build the Talker/Listener application, select this directory for the location, and enter a suitable name for the project, such as TL.
2. Add the following Visual C++ source files to the project:



- `..\GPIBERR.C`

- `..\GPIBREAD`

- `..\GPIBWAIT.C`

- `..\GPIBWRT.C`

1. Add the following source files provided by National Instruments to the project:

- `..\decl-32.h`

- `..gpib-32.obj`

1. Add the appropriate program main source file to the project (see Overview for a list of the three source files).

2. Build and test the project.

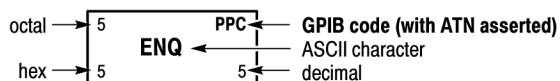
3. To build another of the example projects, repeat steps 4 through 8.



# Appendix A: Character Set

| B7<br>B6<br>B5<br>BITS<br>B4 B3 B2 B1 | 0 0 0                 |           | 0 0 1                 |             | 0 1 0               |              | 0 1 1    |                        | 1 0 0             |          | 1 0 1    |           | 1 1 0                              |          | 1 1 1     |           |
|---------------------------------------|-----------------------|-----------|-----------------------|-------------|---------------------|--------------|----------|------------------------|-------------------|----------|----------|-----------|------------------------------------|----------|-----------|-----------|
|                                       | CONTROL               |           |                       |             | NUMBERS<br>SYMBOLS  |              |          |                        | UPPER CASE        |          |          |           | LOWER CASE                         |          |           |           |
| 0 0 0 0                               | 0<br>NUL              | 20<br>DLE | 40<br>SP              | 60<br>0     | 100<br>@            | 120<br>P     | 140<br>' | 160<br>p               | 0<br>64           | 50<br>80 | 60<br>96 | 70<br>112 |                                    |          |           |           |
| 0 0 0 1                               | 1<br>SOH              | 21<br>DC1 | 41<br>!               | 61<br>1     | 101<br>A            | 121<br>Q     | 141<br>a | 161<br>q               | 1<br>11           | 21<br>17 | 21<br>33 | 31<br>49  | 41<br>65                           | 51<br>81 | 61<br>97  | 71<br>113 |
| 0 0 1 0                               | 2<br>STX              | 22<br>DC2 | 42<br>"               | 62<br>2     | 102<br>B            | 122<br>R     | 142<br>b | 162<br>r               | 2<br>12           | 22<br>18 | 22<br>34 | 32<br>50  | 42<br>66                           | 52<br>82 | 62<br>98  | 72<br>114 |
| 0 0 1 1                               | 3<br>ETX              | 23<br>DC3 | 43<br>#               | 63<br>3     | 103<br>C            | 123<br>S     | 143<br>c | 163<br>s               | 3<br>13           | 23<br>19 | 23<br>35 | 33<br>51  | 43<br>67                           | 53<br>83 | 63<br>99  | 73<br>115 |
| 0 1 0 0                               | 4<br>EOT              | 24<br>DC4 | 44<br>\$              | 64<br>4     | 104<br>D            | 124<br>T     | 144<br>d | 164<br>t               | 4<br>14           | 24<br>20 | 24<br>36 | 34<br>52  | 44<br>68                           | 54<br>84 | 64<br>100 | 74<br>116 |
| 0 1 0 1                               | 5<br>ENQ              | 25<br>NAK | 45<br>%               | 65<br>5     | 105<br>E            | 125<br>U     | 145<br>e | 165<br>u               | 5<br>15           | 25<br>21 | 25<br>37 | 35<br>53  | 45<br>69                           | 55<br>85 | 65<br>101 | 75<br>117 |
| 0 1 1 0                               | 6<br>ACK              | 26<br>SYN | 46<br>&               | 66<br>6     | 106<br>F            | 126<br>V     | 146<br>f | 166<br>v               | 6<br>16           | 26<br>22 | 26<br>38 | 36<br>54  | 46<br>70                           | 56<br>86 | 66<br>102 | 76<br>118 |
| 0 1 1 1                               | 7<br>BEL              | 27<br>ETB | 47<br>'               | 67<br>7     | 107<br>G            | 127<br>W     | 147<br>g | 167<br>w               | 7<br>17           | 27<br>23 | 27<br>39 | 37<br>55  | 47<br>71                           | 57<br>87 | 67<br>103 | 77<br>119 |
| 1 0 0 0                               | 10<br>BS              | 30<br>CAN | 50<br>(               | 70<br>8     | 110<br>H            | 130<br>X     | 150<br>h | 170<br>x               | 8<br>18           | 28<br>24 | 28<br>40 | 38<br>56  | 48<br>72                           | 58<br>88 | 68<br>104 | 78<br>120 |
| 1 0 0 1                               | 11<br>HT              | 31<br>EM  | 51<br>)               | 71<br>9     | 111<br>I            | 131<br>Y     | 151<br>i | 171<br>y               | 9<br>19           | 29<br>25 | 29<br>41 | 39<br>57  | 49<br>73                           | 59<br>89 | 69<br>105 | 79<br>121 |
| 1 0 1 0                               | A<br>LF               | 32<br>SUB | 52<br>*               | 72<br>:     | 112<br>J            | 132<br>Z     | 152<br>j | 172<br>z               | A<br>10           | 1A<br>26 | 2A<br>42 | 3A<br>58  | 4A<br>74                           | 5A<br>90 | 6A<br>106 | 7A<br>122 |
| 1 0 1 1                               | B<br>VT               | 33<br>ESC | 53<br>+               | 73<br>;     | 113<br>K            | 133<br>[     | 153<br>k | 173<br>{               | B<br>11           | 1B<br>27 | 2B<br>43 | 3B<br>59  | 4B<br>75                           | 5B<br>91 | 6B<br>107 | 7B<br>123 |
| 1 1 0 0                               | C<br>FF               | 34<br>FS  | 54<br>,               | 74<br><     | 114<br>L            | 134<br>\<br> | 154<br>l | 174<br>;               | C<br>12           | 1C<br>28 | 2C<br>44 | 3C<br>60  | 4C<br>76                           | 5C<br>92 | 6C<br>108 | 7C<br>124 |
| 1 1 0 1                               | D<br>CR               | 35<br>GS  | 55<br>-               | 75<br>=     | 115<br>M            | 135<br>]<br> | 155<br>m | 175<br>}               | D<br>13           | 1D<br>29 | 2D<br>45 | 3D<br>61  | 4D<br>77                           | 5D<br>93 | 6D<br>109 | 7D<br>125 |
| 1 1 1 0                               | E<br>SO               | 36<br>RS  | 56<br>.               | 76<br>>     | 116<br>N            | 136<br>^     | 156<br>n | 176<br>~               | E<br>14           | 1E<br>30 | 2E<br>46 | 3E<br>62  | 4E<br>78                           | 5E<br>94 | 6E<br>110 | 7E<br>126 |
| 1 1 1 1                               | F<br>SI               | 37<br>US  | 57<br>/<br>           | 77<br>?<br> | 117<br>O            | 137<br>-<br> | 157<br>o | 177<br>RUBOUT<br>(DEL) | F<br>15           | 1F<br>31 | 2F<br>47 | 3F<br>63  | 4F<br>79                           | 5F<br>95 | 6F<br>111 | 7F<br>127 |
|                                       | ADDRESSED<br>COMMANDS |           | UNIVERSAL<br>COMMANDS |             | LISTEN<br>ADDRESSES |              |          |                        | TALK<br>ADDRESSES |          |          |           | SECONDARY ADDRESSES<br>OR COMMANDS |          |           |           |

## KEY



## Tektronix

REF: ANSI STD X3.4-1977  
 IEEE STD 488.1-1987  
 ISO STD 646-2973



## Appendix B: Reserved Words

### Reserved Words

This is a list of reserved words for the DPO4000 Series instruments. Capital letters identify the required minimum spelling. For the most robust code, use the full spelling, since spelling rules may change over time and among instrument models.

|               |               |             |                |
|---------------|---------------|-------------|----------------|
| *CAL          | BUS           | DESTination | HBars          |
| *CLS          | BUSY          | DHCPbootp   | HDR            |
| *DDT          | BY            | DIAG        | HdTv           |
| *ESE          | BYT_Nr        | DIR         | HEADer         |
| *ESR          | BYT_Or        | DIREction   | HIGH           |
| *IDN          | CAL           | DIREction   | HIGHLimit      |
| *LRN          | CALibrate     | DISPLAY     | HOLDTime       |
| *OPC          | CAN           | DISplay     | HOLDoff        |
| *OPT          | CATalog       | DNS         | HORZ           |
| *PSC          | CH            | DOMAINname  | HORizontal     |
| *PUD          | CH2           | DOTsonly    | HORizontal     |
| *RCL          | CH4           | DUE         | HOURS          |
| *RST          | CLAss         | EDGE        | HPOS           |
| *SAV          | CLEAR         | ENCdg       | HTTPPort       |
| *SRE          | CLEARMenu     | END         | HWVersion      |
| *STB          | CLEARSnapshot | ENET        | I2C            |
| *TRG          | CLOCK         | ETHERnet    | ID             |
| *TST          | CLOCK         | EVENT       | IDentifier     |
| *WAI          | CONDition     | EVENTS      | IMAGe          |
| A             | CONTRAST      | EVMsg       | IMMed          |
| ABSolute      | CONTRol       | EVQty       | IN             |
| ACQLENGTH     | COPy          | FACTOR      | INCRement      |
| ACQuire       | COUNT         | FACTory     | INDICators     |
| ACTIVEprinter | COUPling      | FASTAcq     | INKSaver       |
| ADD           | CREATE        | FIELD       | INPut          |
| ADDRESS       | CURSor        | FILEFormat  | INTENSITY      |
| ADDRESS       | CURVe         | FILESystem  | INTERNAL       |
| ALias         | CUSTOM        | FLAG        | INVert         |
| ALias         | CWD           | FOCUS       | IPADDRESS      |
| ALL           | D             | FORCERange  | LANGuage       |
| ALLEV         | DATA          | FORMAT      | LAYout         |
| AUTOContrast  | DATE          | FORMAT      | LESSLimit      |
| AUTOSet       | DATA          | FPANEL      | LEVeL          |
| AUTOZero      | DEFine        | FRAMetype   | LINE           |
| AUXout        | DEFine        | FREE        | LIST           |
| B             | DEGAUSS       | FREESpace   | LOCK           |
| BACKLight     | DELEte        | FUNCTION    | LOG            |
| BANDwidth     | DELEte        | Frequency   | LOGIC          |
| BELl          | DELTA         | GAIN        | LOOP           |
| BITRate       | DELTatime     | GATEway     | LOW            |
| BIT_Nr        | DELaY         | GATING      | LOWLimit       |
| BN_Fmt        | DELeTe        | GPIBUSb     | LOWerthreshold |
| BOX           | DESE          | GRaticule   | MAG            |
| BRightness    | DESKew        | HARDCopy    | MAIn           |

|               |              |              |                |
|---------------|--------------|--------------|----------------|
| MAIn          | PRESS        | SHOW         | TRIGger        |
| MARK          | PREVIEW      | SIGna1       | TURN           |
| MARKSINCOLUMN | PREViewstate | SIZE         | TYPE           |
| MATH          | PRINTER      | SIZE         | TYPE           |
| MATHVAR       | PRObe        | SLEWRate     | TYsetting      |
| MAXSamplerate | PT_Fmt       | SLOpe        | Trigger        |
| MAXimum       | PT_Order     | SNAPShot     | UNITS          |
| MEAN          | PT_off       | SOURCE       | UNITS          |
| MEAS          | PULSEWIDTH   | SOURCE2      | UNLock         |
| MEASurement   | PULSEwidth   | SOURCe       | UPPERthreshold |
| MESSAge       | PULSe        | SPC          | USE            |
| METHod        | QUALifier    | SPECTra1     | VALue          |
| MID           | RECA11       | SPI          | VAR            |
| MID2          | RECOrdlength | SPLit        | VBArs          |
| MINIMUM       | REF          | SS           | VDELTA         |
| MISO          | REFLevel     | STANDARD     | VECTorscope    |
| MKDir         | REM          | START        | VERBoSe        |
| MODE          | REName       | STATE        | VERT           |
| Mode          | RESOLution   | STATISTICS   | VERTical       |
| MORELimit     | RESUlt       | STATUS       | VIDeo          |
| MOSI          | RESistance   | STATE        | Voltage        |
| NAME          | RESUltS      | STATUS       | WAVEform       |
| NEWpass       | RISEFall     | STARTuptime  | WAVFrm         |
| NOTIfy        | RMDir        | STDdev       | WEIGHTing      |
| NR_Pt         | ROLL         | STEPSTATUS   | WFID           |
| NTIMes        | RUNT         | STEPSTIMulus | WFMInpre       |
| NUMACq        | SAMPLERate   | STOP         | WFMOutpre      |
| NUMAVg        | SAMPLEpoint  | STOPAfter    | WHEN           |
| NUMEnv        | SAVE         | STRing       | WIDTH          |
| NUMHORZ       | SCAN         | STYle        | WINDow         |
| NUMVERT       | SCAlE        | STsetting    | XINcr          |
| OFFSet        | SCLK         | SUBNETMask   | XUNit          |
| OPTion        | SDATA        | SYNC         | XY             |
| OUT           | SEARCH       | TEKSecure    | XZerO          |
| OWNeR         | SECdiv       | TEMPerature  | YCHannel       |
| PASSword      | SElect       | TERmination  | YEARS          |
| PATtern       | SElected     | THREShold    | YMu1t          |
| PERCent       | SERnumber    | THREShold    | YOFF           |
| PERSistence   | SET          | TIME         | YREF           |
| PICTure       | SETADDRESS   | TIME         | YUNit          |
| POLARity      | SETHold      | TOTAL        | YUNits         |
| POLarity      | SETID        | TOTAL        | YZerO          |
| POSITION      | SETtime      | TOTALuptime  | ZOOM           |
| POStion       | SETUp        | TRANSition   | ZOOM           |

## Appendix C: Factory Default Setup Values

**### Writer's note: These default setup values need to be updated for the DPO4000. The ones now shown are really just for the TDS5000..** The following tables list the default setup values by command group. The instrument sets these values when you press the front-panel DEFAULT SETUP button or sending the FACTORY command. Only those commands that have values set by the DEFAULT SETUP function are listed in these tables.

**Note:** The following commands do not affect the DATA parameters: TEKSecure, \*RST, RECALL:SETUP FACTORY. These are the initial factory default settings for DATA :DATA:DESTINATION REF1;ENCDG RIBINARY;SOURCE CH1;START 1;STOP 5000;FRAMESTART 1;FRAMESTOP 5000.

### Acquisition Default Values

The following table lists the Acquisition factory default setup values.

| Command              | Default Values |
|----------------------|----------------|
| ACQUIRE:MODE         | SAMPLE         |
| ACQUIRE:NUMAVG       | 16             |
| ACQUIRE:NUMENV       | 10             |
| ACQUIRE:NUMSAMPLES   | 16000          |
| ACQUIRE:SAMPLINGMODE | IT             |
| ACQUIRE:STATE        | 1 (ON)         |
| ACQUIRE:STOPAFTER    | RUNSTOP        |
| FASTACQ:STATE        | 0              |

### Alias Default Values

The following table lists the Alias factory default setup values.

| Command     | Default Values |
|-------------|----------------|
| ALIAS:STATE | 0 (OFF)        |

### Cursor Default Values

The following table lists the Cursor factory default setup values.

| Command                 | Default Values |
|-------------------------|----------------|
| CURSOR:FUNCTION         | VBARS          |
| CURSOR:HBARS:POSITION1  | 300.0000E-3    |
| CURSOR:HBARS:POSITION2  | -300.0000E-3   |
| CURSOR:LINESTYLE        | SOLID          |
| CURSOR:MODE             | INDEPENDENT    |
| CURSOR:PAIRED:POSITION1 | -1.6000E-6     |
| CURSOR:PAIRED:POSITION2 | 1.6000E-6      |

|                           |             |
|---------------------------|-------------|
| CURSOR:SCREEN:STYLE       | LINES       |
| CURSOR:SOURCE1            | CH1         |
| CURSOR:SOURCE2            | CH1         |
| CURSOR:STATE              | 0           |
| CURSOR:VBARS:POSITION1    | -1.6000E-6  |
| CURSOR:VBARS:POSITION2    | 1.6000E-6   |
| CURSOR:VBARS:UNITS        | SECONDS     |
| CURSOR:WAVEFORM:POSITION1 | -1.6000E-6  |
| CURSOR:WAVEFORM:POSITION2 | 1.6000E-6   |
| CURSOR:WAVEFORM:SOURCE2   | CH1         |
| CURSOR:WAVEFORM:STYLE     | LINE_X      |
| CURSOR:WAVEFORM:UNITS     | BASE        |
| CURSOR:XY:READOUT         | RECTANGULAR |
| CURSOR:XY:RECTX1          | 0.0000      |
| CURSOR:XY:RECTX2          | 0.0000      |
| CURSOR:XY:RECTY1          | 0.0000      |
| CURSOR:XY:RECTY2          | 0.0000      |

#### Diagnostics Default Values

The following table lists the Diagnostics factory default setup values.

| Command    | Default Values |
|------------|----------------|
| DIAG:LEVEL | SUBSYS         |

#### Display Default Values

The following table lists the Display factory default setup values.

| Command                              | Default Values |
|--------------------------------------|----------------|
| DISPLAY:CLOCK                        | 1              |
| DISPLAY:COLOR:MATHCOLOR              | DEFAULT        |
| DISPLAY:COLOR:PALETTE:IMAGEVIEW      | TEMPERATURE    |
| DISPLAY:COLOR:PALETTE:RECORDVIEW     | NORMAL         |
| DISPLAY:COLOR:PALETTE:USER:CARET     | 150,50,100     |
| DISPLAY:COLOR:PALETTE:USER:CH1       | 180,50,100     |
| DISPLAY:COLOR:PALETTE:USER:CH2       | 300,50,100     |
| DISPLAY:COLOR:PALETTE:USER:CH3       | 60,50,100      |
| DISPLAY:COLOR:PALETTE:USER:CH4       | 240,50,100     |
| DISPLAY:COLOR:PALETTE:USER:GRATICULE | 165,50,15      |
| DISPLAY:COLOR:PALETTE:USER:HISTOGRAM | 320,50,100     |
| DISPLAY:COLOR:PALETTE:USER:MASK      | 0,25,75        |



|  |            |
|--|------------|
| DISPLAY:COLOR:PALETTE:USER:MASKHIGHLIGHT | 160,50,100 |
| DISPLAY:COLOR:PALETTE:USER:MATH1         | 160,50,100 |
| DISPLAY:COLOR:PALETTE:USER:MATH2         | 40,60,100  |
| DISPLAY:COLOR:PALETTE:USER:MATH3         | 120,60,100 |
| DISPLAY:COLOR:PALETTE:USER:MATH4         | 195,50,100 |
| DISPLAY:COLOR:PALETTE:USER:REF1          | 0,90,0     |
| DISPLAY:COLOR:PALETTE:USER:REF2          | 0,90,100   |
| DISPLAY:COLOR:PALETTE:USER:REF3          | 60,90,100  |
| DISPLAY:COLOR:PALETTE:USER:REF4          | 240,90,100 |
| DISPLAY:COLOR:REFCOLOR                   | DEFAULT    |
| DISPLAY:CUSORTICK                        | SHORT      |
| DISPLAY:FILTER                           | SINX       |
| DISPLAY:FORMAT                           | YT         |
| DISPLAY:GRATICULE                        | FULL       |
| DISPLAY:INTENSITY:AUTOBRIGHT             | 1          |
| DISPLAY:INTENSITY:SCREENSAVER            | 1          |
| DISPLAY:INTENSITY:SCREENSAVERDELAY       | 28800      |
| DISPLAY:INTENSITY:WAVEFORM:IMAGEVIEW     | 75.0000    |
| DISPLAY:INTENSITY:WAVEFORM:RECORDVIEW    | 75.0000    |
| DISPLAY:PERSISTENCE                      | OFF        |
| DISPLAY:SCREENTEXT:LABEL1:NAME           | ""         |
| DISPLAY:SCREENTEXT:LABEL1:XPOS           | 100        |
| DISPLAY:SCREENTEXT:LABEL1:YPOS           | 5          |
| DISPLAY:SCREENTEXT:LABEL2:NAME           | ""         |
| DISPLAY:SCREENTEXT:LABEL2:XPOS           | 100        |
| DISPLAY:SCREENTEXT:LABEL2:YPOS           | 20         |
| DISPLAY:SCREENTEXT:LABEL3:NAME           | ""         |
| DISPLAY:SCREENTEXT:LABEL3:XPOS           | 100        |
| DISPLAY:SCREENTEXT:LABEL3:YPOS           | 35         |
| DISPLAY:SCREENTEXT:LABEL4:NAME           | ""         |
| DISPLAY:SCREENTEXT:LABEL4:XPOS           | 100        |
| DISPLAY:SCREENTEXT:LABEL4:YPOS           | 5          |
| DISPLAY:SCREENTEXT:LABEL5:NAME           | ""         |
| DISPLAY:SCREENTEXT:LABEL5:XPOS           | 100        |
| DISPLAY:SCREENTEXT:LABEL5:YPOS           | 65         |
| DISPLAY:SCREENTEXT:LABEL6:NAME           | ""         |
| DISPLAY:SCREENTEXT:LABEL6:XPOS           | 100        |
| DISPLAY:SCREENTEXT:LABEL6:YPOS           | 80         |
| DISPLAY:SCREENTEXT:LABEL7:NAME           | ""         |

|                                |             |
|--------------------------------|-------------|
| DISPLAY:SCREENTEXT:LABEL7:XPOS | 100         |
| DISPLAY:SCREENTEXT:LABEL7:YPOS | 95          |
| DISPLAY:SCREENTEXT:LABEL8:NAME | ""          |
| DISPLAY:SCREENTEXT:LABEL8:XPOS | 100         |
| DISPLAY:SCREENTEXT:LABEL8:YPOS | 110         |
| DISPLAY:SCREENTEXT:STATE       | 0           |
| DISPLAY:STYLE                  | VECTORS     |
| DISPLAY:TRIGBAR                | SHORT       |
| DISPLAY:TRIGT                  | 1           |
| DISPLAY:VARPERSIST             | 500.0000E-3 |

#### Email Default Values

The following table lists the Email setup values.

| Command           | Default Values                        |
|-------------------|---------------------------------------|
| EMAIL:ATTEMPTS    | 1                                     |
| EMAIL:AUTHLOGIN   | ""                                    |
| EMAIL:FROM        | "TDS5054B_<Instrument Serial Number>" |
| EMAIL:HOSTWANTED  | ""                                    |
| EMAIL:IMAGE       | 0                                     |
| EMAIL:LIMIT       | 0                                     |
| EMAIL:MASK        | 0                                     |
| EMAIL:MAXSIZE     | 1                                     |
| EMAIL:MEASUREMENT | 0                                     |
| EMAIL:NUMEMAILS   | 1                                     |
| EMAIL:SMTPPORT    | 25                                    |
| EMAIL:SMTPSERVER  | ""                                    |
| EMAIL:TIMEOUT     | 30                                    |
| EMAIL:TO          | ""                                    |
| EMAIL:TRIGGER     | 0                                     |
| EMAIL:WAVEFORM    | 0                                     |

#### Hard Copy Default Values

The following table lists the Hard Copy factory default setup values.

| Command           | Default Values |
|-------------------|----------------|
| HARDCopy:FILENAME | ""             |
| HARDCopy:PORT     | FILE           |

#### Histogram Default Values

The following table lists the Histogram factory default setup values.

| Command            | Default Values                  |
|--------------------|---------------------------------|
| HISTOGRAM:BOXPCNT  | 30.0000,25.1000,70.0000,75.2000 |
| HISTOGRAM:DISPLAY  | LINEAR                          |
| HISTOGRAM:FUNCTION | HORIZONTAL                      |
| HISTOGRAM:SIZE     | 2.0000                          |
| HISTOGRAM:SOURCE   | CH1                             |
| HISTOGRAM:STATE    | 0                               |

#### Horizontal Default Values

The following table lists the Horizontal factory default setup values.

| Command  | Default Values |
|--|----------------|
| HORIZONTAL:DELAY:MODE  | 0              |
| HORIZONTAL:DELAY:POSITION                                      | 50.0000        |
| HORIZONTAL:DELAY:TIME  | 0.000000000000 |
| HORIZONTAL:FASTFRAME:COUNT                                     | 2              |
| HORIZONTAL:FASTFRAME:LENGTH                                    | 5000           |
| HORIZONTAL:FASTFRAME:MULTIPLEFRAMES 1<br>:FRAMESTART:CH<1-4>   |                |
| HORIZONTAL:FASTFRAME:MULTIPLEFRAMES 1<br>:FRAMESTART:MATH<1-4> |                |
| HORIZONTAL:FASTFRAME:MULTIPLEFRAMES 1<br>:FRAMESTART:REF<1-4>  |                |
| HORIZONTAL:FASTFRAME:MULTIPLEFRAMES OFF<br>:MODE               |                |
| HORIZONTAL:FASTFRAME:MULTIPLEFRAMES 2<br>:NUMFRAMES:CH1-4>     |                |
| HORIZONTAL:FASTFRAME:MULTIPLEFRAMES 2<br>:NUMFRAMES:MATH<1-4>  |                |
| HORIZONTAL:FASTFRAME:MULTIPLEFRAMES 2<br>:NUMFRAMES:REF<1-4>   |                |
| HORIZONTAL:FASTFRAME:READOUTS                                  | 0              |
| HORIZONTAL:FASTFRAME:REF:FRAME                                 | 1              |
| HORIZONTAL:FASTFRAME:REF:SOURCE                                | CH1            |
| HORIZONTAL:FASTFRAME:SELECTED:CH1                              | 2              |
| HORIZONTAL:FASTFRAME:SELECTED:CH2                              | 2              |
| HORIZONTAL:FASTFRAME:SELECTED:CH3                              | 2              |
| HORIZONTAL:FASTFRAME:SELECTED:CH4                              | 2              |
| HORIZONTAL:FASTFRAME:SELECTED:MATH1                            | 2              |
| HORIZONTAL:FASTFRAME:SELECTED:MATH2                            | 2              |
| HORIZONTAL:FASTFRAME:SELECTED:MATH3                            | 2              |

|                                     |             |
|-------------------------------------|-------------|
| HORIZONTAL:FASTFRAME:SELECTED:MATH4 | 2           |
| HORIZONTAL:FASTFRAME:SELECTED:REF1  | 2           |
| HORIZONTAL:FASTFRAME:SELECTED:REF2  | 2           |
| HORIZONTAL:FASTFRAME:SELECTED:REF3  | 2           |
| HORIZONTAL:FASTFRAME:SELECTED:REF4  | 2           |
| HORIZONTAL:FASTFRAME:STATE          | 0           |
| HORIZONTAL:FASTFRAME:SUMFRAME       | NONE        |
| HORIZONTAL:FASTFRAME:TRACK          | LIVE        |
| HORIZONTAL:MAIN:POSITION            | 50.0000     |
| HORIZONTAL:MAIN:SAMPLERATE          | 1.2500E+9   |
| HORIZONTAL:MAIN:SCALE               | 400.0000E-9 |
| HORIZONTAL:RECORDLENGTH             | 5000        |
| HORIZONTAL:RESOLUTION               | 5000        |
| HORIZONTAL:ROLL                     | AUTO        |

#### Limit Test Default Values

The following table lists the Limit Test factory default setup values.

| Command                             | Default Values |
|-------------------------------------|----------------|
| LIMIT:BEEP                          | 0              |
| LIMIT:COMPARE:<CH1-4>               | NONE           |
| LIMIT:COMPARE:MATH<1-4>             | NONE           |
| LIMIT:COMPARE:REF<1-4>              | NONE           |
| LIMIT:EMAIL                         | 0              |
| LIMIT:HARDCOPY                      | 0              |
| LIMIT:HIGHLIGHTHITS                 | 1              |
| LIMIT:LOCK                          | 1              |
| LIMIT:LOG                           | 0              |
| LIMIT:SAVEWFM                       | 0              |
| LIMIT:SRQ                           | 0              |
| LIMIT:STATE                         | 0              |
| LIMIT:STOPONVIOLATION               | 0              |
| LIMIT:TEMPLATE:TOLERANCE:HORIZONTAL | 40.0000E-3     |
| LIMIT:TEMPLATE:TOLERANCE:VERTICAL   | 40.0000E-3     |

#### Mask Default Values

The following table lists the Mask factory default setup values.

| Command         | Default Values |
|-----------------|----------------|
| MASK:AUTOADJUST | 10             |

|                           |             |
|---------------------------|-------------|
| MASK:AUTOADJUST:HDELTA    | 10          |
| MASK:AUTOADJUST:VDELTA    | 10          |
| MASK:AUTOSET:AUTOADJUST   | 1           |
| MASK:AUTOSET:HPOS         | 1           |
| MASK:AUTOSET:HSCALE       | 1           |
| MASK:AUTOSET:MODE         | MANUAL      |
| MASK:AUTOSET:OFFSETADJ    | 1           |
| MASK:AUTOSET:STANDARD     | NONE        |
| MASK:AUTOSET:TRIGGER      | 1           |
| MASK:AUTOSET:VPOS         | 1           |
| MASK:AUTOSET:VSCALE       | 1           |
| MASK:COUNT:STATE          | 0           |
| MASK:DISPLAY              | 1           |
| MASK:FILTER               | 0           |
| MASK:HIGHLIGHTHITS        | 1           |
| MASK:INVERT               | 0           |
| MASK:LOCK                 | 1           |
| MASK:MARGIN:PERCENT       | 5.0000      |
| MASK:MARGIN:STATE         | 0           |
| MASK:MASKPRE:AMPLITUDE    | 100.0000E-3 |
| MASK:MASKPRE:HSCALE       | 200.0000E-9 |
| MASK:MASKPRE:HTRIGPOS     | 500.0000E-3 |
| MASK:MASKPRE:PATTERNBITS  | 1           |
| MASK:MASKPRE:PRESAMPBITS  | 0           |
| MASK:MASKPRE:RECORDLENGTH | 5000        |
| MASK:MASKPRE:TRIGTOSAMP   | 0.0000      |
| MASK:MASKPRE:VOFFSET      | 0.0000      |
| MASK:MASKPRE:VPOS         | 0.0000      |
| MASK:MASKPRE:VSCALE       | 1.0000E-3   |
| MASK:MASKPRE:WIDTH        | 0.0000      |
| MASK:POLARITY             | POSITIVE    |
| MASK:SOURCE               | CH1         |
| MASK:STANDARD             | NONE        |
| MASK:STOPONVIOLATION      | 0           |
| MASK:TEST:BEEP:COMPLETION | 0           |
| MASK:TEST:BEEP:FAILURE    | 0           |
| MASK:TEST:DELAY           | 0.0000      |
| MASK:TEST:HARDCOPY        | 0           |
| MASK:TEST:LOG:FAILURE     | 0           |

|                            |                          |
|----------------------------|--------------------------|
| MASK:TEST:REPEAT           | 0                        |
| MASK:TEST:SAMPLE:THRESHOLD | 1                        |
| MASK:TEST:SAVEWFM          | 0                        |
| MASK:TEST:SAVEWFM:FILENAME | "C:/TekScope/Waveforms/" |
| MASK:TEST:SRQ:COMPLETION   | 0                        |
| MASK:TEST:SRQ:FAILURE      | 0                        |
| MASK:TEST:STATE            | 0                        |
| MASK:TEST:STOP:FAILURE     | 0                        |
| MASK:TEST:THRESHOLD        | 1                        |
| MASK:TEST:WAVEFORM         | 20                       |
| MASK:USER:AMPLITUDE        | 100.0000E-3              |
| MASK:USER:BITRATE          | 1544000                  |
| MASK:USER:HSCALE           | 200.0000E-9              |
| MASK:USER:HTRIGPOS         | 500.0000E-3              |
| MASK:USER:LABEL            | "User Mask"              |
| MASK:USER:PATTERNBITS      | 1                        |
| MASK:USER:PRESAMPBITS      | 0                        |
| MASK:USER:RECORDLENGTH     | 5000                     |
| MASK:USER:TRIGTOSAMP       | 0.0000                   |
| MASK:USER:VOFFSET          | 0.0000                   |
| MASK:USER:VPOS             | 0.0000                   |
| MASK:USER:VSCALE           | 1.0000E-3                |
| MASK:USER:WIDTH            | 0.0000                   |

#### Math Default Values

The following table lists the Math factory default setup values.

| Command                      | Default Values   |
|------------------------------|------------------|
| MATH<1-4>:DEFINE             | " "              |
| MATH<1-4>:LABEL:NAME         | ""               |
| MATH<1-4>:LABEL:XPOS         | 1.0000           |
| MATH<1-4>:LABEL:YPOS         | <1> 600.0000E-3  |
|                              | <2> 200.0000E-3  |
|                              | <3> -200.0000E-3 |
|                              | <4> -600.0000E-3 |
| MATH<1-4>::NUMAVG            | 2                |
| MATH<1-4>:SPECTRAL:CENTER    | 312.5000E+6      |
| MATH<1-4>:SPECTRAL:GATEPOS   | 0.0000           |
| MATH<1-4>:SPECTRAL:GATEWIDTH | 2.0000E-6        |
| MATH<1-4>:SPECTRAL:LOCK      | 0                |

|                                   |              |
|-----------------------------------|--------------|
| MATH<1-4>:SPECTRAL:MAG            | DB           |
| MATH<1-4>:SPECTRAL:PHASE          | DEGREES      |
| MATH<1-4>:SPECTRAL:REFLEVEL       | 20.0000      |
| MATH<1-4>:SPECTRAL:REFLEVELOFFSET | 223.6000E-3  |
| MATH<1-4>:SPECTRAL:RESBW          | 1000.0000E+3 |
| MATH<1-4>:SPECTRAL:SPAN           | 625.0000E+6  |
| MATH<1-4>:SPECTRAL:SUPPRESS       | -200.0000    |
| MATH<1-4>:SPECTRAL:UNWRAP         | 0            |
| MATH<1-4>:SPECTRAL:WINDOW         | GAUSSIAN     |
| MATH<1-4>:UNITSTRING              | "V"          |
| MATH<1-4>:VERTICAL:POSITION       | 0.0000       |
| MATH<1-4>:VERTICAL:SCALE          | 1.0000       |
| MATHVAR:VAR<1-8>                  | 0.0000       |

### Measurement Default Values

The following table lists the Measurement factory default setup values.

| Command                               | Default Values |
|---------------------------------------|----------------|
| MEASUREMENT:GATING                    | OFF            |
| MEASUREMENT:IMMED:DELAY:DIRECTION     | FORWARDS       |
| MEASUREMENT:IMMED:DELAY:EDGE1         | RISE           |
| MEASUREMENT:IMMED:DELAY:EDGE2         | RISE           |
| MEASUREMENT:IMMED:METHOD              | HISTOGRAM      |
| MEASUREMENT:IMMED:NOISE               | HIGH           |
| MEASUREMENT:IMMED:REFLEVEL:ABSOLUTE   | 0.0000         |
| MEASUREMENT:IMMED:REFLEVEL:ABSOLUTE   | 0.0000         |
| MEASUREMENT:IMMED:REFLEVEL:ABSOLUTE   | 0.0000         |
| MEASUREMENT:IMMED:REFLEVEL:ABSOLUTE   | 0.0000         |
| MEASUREMENT:IMMED:REFLEVEL:METHOD     | PERCENT        |
| MEASUREMENT:IMMED:REFLEVEL:PERCENT    | 90.0000        |
| MEASUREMENT:IMMED:REFLEVEL:PERCENT    | 10.0000        |
| MEASUREMENT:IMMED:REFLEVEL:PERCENT    | 50.0000        |
| MEASUREMENT:IMMED:REFLEVEL:PERCENT    | 50.0000        |
| MEASUREMENT:IMMED:SOURCE1             | CH1            |
| MEASUREMENT:IMMED:SOURCE1:SIGTYPE     | PULSE          |
| MEASUREMENT:IMMED:SOURCE2             | CH1            |
| MEASUREMENT:IMMED:SOURCE2:SIGTYPE     | PULSE          |
| MEASUREMENT:IMMED:TYPE                | UNDEFINED      |
| MEASUREMENT:MEAS<1-8>:DELAY:DIRECTION | FORWARDS       |
| MEASUREMENT:MEAS<1-8>:DELAY:EDGE1     | RISE           |

|   |           |
|---|-----------|
| MEASUREMENT:MEAS<1-8>:DELAY:EDGE2       | RISE      |
| MEASUREMENT:MEAS<1-8>:METHOD            | HISTOGRAM |
| MEASUREMENT:MEAS<1-8>:NOISE             | HIGH      |
| MEASUREMENT:MEAS<1-8>:REFLEVEL:ABSOLUTE | HIGH      |
| MEASUREMENT:MEAS<1-8>:REFLEVEL:ABSOLUTE | LOW       |
| MEASUREMENT:MEAS<1-8>:REFLEVEL:ABSOLUTE | MID1      |
| MEASUREMENT:MEAS<1-8>:REFLEVEL:ABSOLUTE | MID2      |
| MEASUREMENT:MEAS<1-8>:REFLEVEL:METH     | PERCENT   |
| MEASUREMENT:MEAS<1-8>:REFLEVEL:PERCENT  | HIGH9     |
| MEASUREMENT:MEAS<1-8>:REFLEVEL:PERCENT  | LOW1      |
| MEASUREMENT:MEAS<1-8>:REFLEVEL:PERCENT  | MID15     |
| MEASUREMENT:MEAS<1-8>:REFLEVEL:PERCENT  | MID2      |
| MEASUREMENT:MEAS<1-8>:SOURCE1           | CH1       |
| MEASUREMENT:MEAS<1-8>:SOURCE1:SIGTYPE   | PULSE     |
| MEASUREMENT:MEAS<1-8>:SOURCE2           | CH1       |
| MEASUREMENT:MEAS<1-8>:SOURCE2:SIGTYPE   | PULSE     |
| MEASUREMENT:MEAS<1-8>:STATE             | 0         |
| MEASUREMENT:MEAS<1-8>:TYPE              | UNDEFINED |
| MEASUREMENT:STATISTICS:MODE             | ALL       |
| MEASUREMENT:STATISTICS:WEIGHTING        | 32        |

#### Miscellaneous Default Values

The following table lists the Miscellaneous factory default setup values.

| Command       | Default Values          |
|---------------|-------------------------|
| ALIAS:STATE   | 0 (OFF)                 |
| AUXOUT:EDGE   | FALLING <cannot change> |
| AUXOUT:SOURCE | ATRIGGER                |
| CMDBATCH      | 1 (ON)                  |
| HEADER        | 1 (ON)                  |
| LOCK          | NONE                    |
| ROSC:SOURCE   | INTERNAL                |
| VERBOSE       | 1 (ON)                  |

#### Save and Recall Default Settings

The following table lists the Save and Recall factory default setup values.

| Command                  | Default Values |
|--------------------------|----------------|
| SAVE:WAVEFORM:FILEFORMAT | INTERNAL       |

#### Trigger Default Values



The following table lists the Trigger factory default setup values.

| Command                                 | Default Values |
|---|----------------|
| TRIGGER:A:COMMUNICATION:AMI:PULSEFORM   | PULSE          |
| TRIGGER:A:COMMUNICATION:AMI:THRESHOLD   | 10000H         |
| TRIGGER:A:COMMUNICATION:AMI:THRESHOLD   | 10000W         |
| TRIGGER:A:COMMUNICATION:BITRATE         | 1544000        |
| TRIGGER:A:COMMUNICATION:CLOCK:POLARITY  | RISE           |
| TRIGGER:A:COMMUNICATION:CMI:PULSEFORM   | PULSE          |
| TRIGGER:A:COMMUNICATION:CODE            | NRZ            |
| TRIGGER:A:COMMUNICATION:SOURCE          | CH1            |
| TRIGGER:A:COMMUNICATION:SOURCE:TYPE     | DATA           |
| TRIGGER:A:COMMUNICATION:STANDARD        | CUSTOM         |
| TRIGGER:A:EDGE:COUPLING                 | DC             |
| TRIGGER:A:EDGE:SLOPE                    | RISE           |
| TRIGGER:A:EDGE:SOURCE                   | CH1            |
| TRIGGER:A:HOLDOFF:BY                    | DEFAULT        |
| TRIGGER:A:HOLDOFF:TIME                  | 1.5000E-6      |
| TRIGGER:A:LEVEL                         | 0.0000         |
| TRIGGER:A:LOGIC:CLASS                   | PATTERN        |
| TRIGGER:A:LOGIC:FUNCTION                | AND            |
| TRIGGER:A:LOGIC:INPUT:CH1               | HIGH           |
| TRIGGER:A:LOGIC:INPUT:CH2               | X              |
| TRIGGER:A:LOGIC:INPUT:CH3               | X              |
| TRIGGER:A:LOGIC:PATTERN:INPUT:CH4       | X              |
| TRIGGER:A:LOGIC:PATTERN:WHEN            | TRUE           |
| TRIGGER:A:LOGIC:PATTERN:WHEN:LESSLIMIT  | 5.0000E-9      |
| TRIGGER:A:LOGIC:PATTERN:WHEN:MORELIMIT  | 5.0000E-9      |
| TRIGGER:A:LOGIC:SETHOLD:CLOCK:EDGE      | RISE           |
| TRIGGER:A:LOGIC:SETHOLD:CLOCK:SOURCE    | CH2            |
| TRIGGER:A:LOGIC:SETHOLD:CLOCK:THRESHOLD | 0.2000         |
| TRIGGER:A:LOGIC:SETHOLD:DATA:SOURCE     | CH1            |
| TRIGGER:A:LOGIC:SETHOLD:DATA:THRESHOLD  | 0.2000         |
| TRIGGER:A:LOGIC:SETHOLD:HOLDTIME        | 2.0000E-9      |
| TRIGGER:A:LOGIC:SETHOLD:SETTIME         | 3.0000E-9      |
| TRIGGER:A:LOGIC:STATE:INPUT:CH4         | RISE           |
| TRIGGER:A:LOGIC:STATE:WHEN              | TRUE           |
| TRIGGER:A:LOGIC:THRESHOLD:<CH1-4>       | 1.2000         |
| TRIGGER:A:MODE                          | AUTO           |
| TRIGGER:A:PULSE:CLASS                   | GLITCH         |

|  |             |
|--|-------------|
| TRIGGER:A:PULSE:GLITCH:POLARITY                | POSITIVE    |
| TRIGGER:A:PULSE:GLITCH:TRIGIF                  | ACCEPT      |
| TRIGGER:A:PULSE:GLITCH:WIDTH                   | 2.0000E-9   |
| TRIGGER:A:PULSE:RUNT:LOGIC:INPUT:CH<1-4>       | HIGH        |
| TRIGGER:A:PULSE:RUNT:LOGIC:THRESHOLD:CH<1-4>   | 0.2000      |
| TRIGGER:A:PULSE:RUNT:POLARITY                  | POSITIVE    |
| TRIGGER:A:PULSE:RUNT:THRESHOLD:HIGH            | 1.2000      |
| TRIGGER:A:PULSE:RUNT:THRESHOLD:LOW             | 800.0000E-3 |
| TRIGGER:A:PULSE:RUNT:WHEN                      | OCCURS      |
| TRIGGER:A:PULSE:RUNT:WIDTH                     | 2.0000E-9   |
| TRIGGER:A:PULSE:SOURCE                         | CH1         |
| TRIGGER:A:PULSE:TIMEOUT:POLARITY               | STAYSHIGH   |
| TRIGGER:A:PULSE:TIMEOUT:TIME                   | 2.0000E-9   |
| TRIGGER:A:PULSE:TRANSITION:DELTATIME           | 2.0000E-9   |
| TRIGGER:A:PULSE:TRANSITION:POLARITY            | POSITIVE    |
| TRIGGER:A:PULSE:TRANSITION:THRESHOLD:HIGH      | 1.2000      |
| TRIGGER:A:PULSE:TRANSITION:THRESHOLD:LOW       | 800.0000E-3 |
| TRIGGER:A:PULSE:TRANSITION:WHEN                | FASTER THAN |
| TRIGGER:A:PULSE:WIDTH:HIGHLIMIT                | 2.0000E-9   |
| TRIGGER:A:PULSE:WIDTH:LOWLIMIT                 | 2.0000E-9   |
| TRIGGER:A:PULSE:WIDTH:POLARITY                 | POSITIVE    |
| TRIGGER:A:PULSE:WIDTH:WHEN                     | WITHIN      |
| TRIGGER:A:PULSE:WINDOW:LOGIC:INPUT:CH<1-4>     | HIGH        |
| TRIGGER:A:PULSE:WINDOW:LOGIC:THRESHOLD:CH<1-4> | 0.2000      |
| TRIGGER:A:PULSE:WINDOW:THRESHOLD:HIGH          | 1.2000      |
| TRIGGER:A:PULSE:WINDOW:THRESHOLD:LOW           | 800.0000E-3 |
| TRIGGER:A:PULSE:WINDOW:TYPE                    | INSIDE      |
| TRIGGER:A:PULSE:WINDOW:WHEN                    | OCCURS      |
| TRIGGER:A:PULSE:WINDOW:WIDTH                   | 2.0000E-9   |
| TRIGGER:A:TYPE                                 | EDGE        |
| TRIGGER:A:VIDEO:CUSTOM:FORMAT                  | INTERLACED  |
| TRIGGER:A:VIDEO:CUSTOM:SCAN                    | RATE1       |
| TRIGGER:A:VIDEO:FIELD                          | ALLFIELDS   |
| TRIGGER:A:VIDEO:HOLDOFF:FIELD                  | 0.0000      |
| TRIGGER:A:VIDEO:LINE                           | 1           |
| TRIGGER:A:VIDEO:POLARITY                       | NORMAL      |
| TRIGGER:A:VIDEO:SCAN                           | RATE1       |
| TRIGGER:A:VIDEO:SOURCE                         | CH1         |
| TRIGGER:A:VIDEO:STANDARD                       | NTSC        |

|                         |            |
|-------------------------|------------|
| TRIGGER:B:BY            | EVENTS     |
| TRIGGER:B:EDGE:COUPLING | DC         |
| TRIGGER:B:EDGE:SLOPE    | RISE       |
| TRIGGER:B:EDGE:SOURCE   | CH1        |
| TRIGGER:B:EVENTS:COUNT  | 2          |
| TRIGGER:B:LEVEL         | 0.0000     |
| TRIGGER:B:STATE         | 0          |
| TRIGGER:B:TIME          | 16.0000E-9 |
| TRIGGER:B:TYPE          | EDGE       |

### Vertical Default Values

The following table lists the Vertical factory default setup values.

| Command                    | Default Values |
|----------------------------|----------------|
| CH<1-4>:BANDWIDTH          | 500.0000E+6    |
| CH<1-4>:COUPLING           | DC             |
| CH<1-4>:DESKEW             | 0.0000         |
| CH<1-4>:INVERT             | 0              |
| CH<1-4>:LABEL:NAME         | ""             |
| CH<1-4>:LABEL:XPOS         | 1.0000         |
| CH<1-4>:LABEL:YPOS         | <1> 2.2000     |
|                            | <2> 1.8000     |
|                            | <3> 1.4000     |
|                            | <4> 1.000      |
| CH<1-4>:OFFSET             | 0.0000         |
| CH<1-4>:POSITION           | 0.0000         |
| CH<1-4>:PROBEFUNC:EXTATTEN | 1.0000         |
| CH<1-4>:PROBEFUNC:EXTUNITS | "V"            |
| CH<1-4>:SCALE              | 100.0000E-3    |
| CH<1-4>:TERMINATION        | 1.0000E+6      |
| DATA:DESTINATION           | REF1           |
| DATA:ENCDG                 | RIBINARY       |
| DATA:FRAMESTART            | 1              |
| DATA:FRAMESTOP             | 5000           |
| DATA:SOURCE                | CH1            |
| DATA:START                 | 1              |
| DATA:STOP                  | 5000           |
| REF1:LABEL:NAME            | ""             |
| REF1:LABEL:XPOS            | 1.0000         |
| REF1:LABEL:YPOS            | -1.0000        |

|                  |         |
|------------------|---------|
| REF2:LABEL:NAME  | ""      |
| REF2:LABEL:XPOS  | 1.0000  |
| REF2:LABEL:YPOS  | -1.4000 |
| REF3:LABEL:NAME  | ""      |
| REF3:LABEL:XPOS  | 1.0000  |
| REF3:LABEL:YPOS  | -1.8000 |
| REF4:LABEL:NAME  | ""      |
| REF4:LABEL:XPOS  | 1.0000  |
| REF4:LABEL:YPOS  | -2.2000 |
| SELECT:CH1       | 1       |
| SELECT:CH2       | 0       |
| SELECT:CH3       | 0       |
| SELECT:CH4       | 0       |
| SELECT:CONTROL   | CH1     |
| SELECT:MATH<1-4> | 0       |
| SELECT:REF<1-4>  | 0       |

#### Waveform Default Values

The following table lists the Waveform factory default setup values.

| Command          | Default Values |
|------------------|----------------|
| DATA:DESTINATION | REF1           |
| DATA:ENCDG       | RIBINARY       |
| DATA:FRAMESTART  | 1              |
| DATA:FRAMESTOP   | 5000           |
| DATA:SOURCE      | CH1            |
| DATA:START       | 1              |
| DATA:STOP        | 5000           |

#### Zoom Default Values

The following table lists the Zoom factory default setup values.

| Command                                | Default Values |
|--|----------------|
| ZOOM:GRATICULE:SIZE                    | 80             |
| ZOOM:GRATICULE:SPLIT                   | EIGHTYTWENTY   |
| ZOOM:MODE                              | 0              |
| ZOOM:SCROLL:DIRECTION                  | STOP           |
| ZOOM:SCROLL:LOCK                       | 0              |
| ZOOM:SCROLL:SPEED                      | 1              |
| ZOOM:ZOOM1:CH<1-4>:HORIZONTAL:POSITION | 50.0000        |

|  |        |
|--|--------|
| ZOOM:ZOOM1:CH<1-4>:HORIZONTAL:SCALE      | 5      |
| ZOOM:ZOOM1:CH<1-4>:VERTICAL:POSITION     | 0.0000 |
| ZOOM:ZOOM1:CH<1-4>:VERTICAL:SCALE        | 1.0000 |
| ZOOM:ZOOM1:MATH<1-4>:HORIZONTAL:POSITION | 500000 |
| ZOOM:ZOOM1:MATH<1-4>:HORIZONTAL:SCALE    | 5      |
| ZOOM:ZOOM1:MATH<1-4>:VERTICAL:POSITION   | 0.0000 |
| ZOOM:ZOOM1:MATH<1-4>:VERTICAL:SCALE      | 1.0000 |
| ZOOM:ZOOM1:REF<1-4>:HORIZONTAL:POSITION  | 600000 |
| ZOOM:ZOOM1:REF<1-4>:HORIZONTAL:SCALE     | 5      |
| ZOOM:ZOOM1:REF<1-4>:VERTICAL:POSITION    | 0.0000 |
| ZOOM:ZOOM1:REF<1-4>:VERTICAL:SCALE       | 1.0000 |
| ZOOM:ZOOM1:SCROLLLOCK                    | 1      |
| ZOOM:ZOOM1:STATE                         | 1      |
| ZOOM:ZOOM2:CH<1-4>:HORIZONTAL:POSITION   | 0.0000 |
| ZOOM:ZOOM2:CH<1-4>:HORIZONTAL:SCALE      | 5      |
| ZOOM:ZOOM2:CH<1-4>:VERTICAL:POSITION     | 0.0000 |
| ZOOM:ZOOM2:CH<1-4>:VERTICAL:SCALE        | 1.0000 |
| ZOOM:ZOOM2:MATH<1-4>:HORIZONTAL:POSITION | 300000 |
| ZOOM:ZOOM2:MATH<1-4>:HORIZONTAL:SCALE    | 5      |
| ZOOM:ZOOM2:MATH<1-4>:VERTICAL:POSITION   | 0.0000 |
| ZOOM:ZOOM2:MATH<1-4>:VERTICAL:SCALE      | 1.0000 |
| ZOOM:ZOOM2:REF<1-4>:HORIZONTAL:POSITION  | 0.0000 |
| ZOOM:ZOOM2:REF<1-4>:HORIZONTAL:SCALE     | 5      |
| ZOOM:ZOOM2:REF<1-4>:VERTICAL:POSITION    | 0.0000 |
| ZOOM:ZOOM2:REF<1-4>:VERTICAL:SCALE       | 1.0000 |
| ZOOM:ZOOM2:SCROLLLOCK                    | 1      |
| ZOOM:ZOOM2:STATE                         | 0      |
| ZOOM:ZOOM3:CH<1-4>:HORIZONTAL:POSITION   | 0.0000 |
| ZOOM:ZOOM3:CH<1-4>:HORIZONTAL:SCALE      | 5      |
| ZOOM:ZOOM3:CH<1-4>:VERTICAL:POSITION     | 0.0000 |
| ZOOM:ZOOM3:CH<1-4>:VERTICAL:SCALE        | 1.0000 |
| ZOOM:ZOOM3:MATH<1-4>:HORIZONTAL:POSITION | 700000 |
| ZOOM:ZOOM3:MATH<1-4>:HORIZONTAL:SCALE    | 5      |
| ZOOM:ZOOM3:MATH<1-4>:VERTICAL:POSITION   | 0.0000 |
| ZOOM:ZOOM3:MATH<1-4>:VERTICAL:SCALE      | 1.0000 |
| ZOOM:ZOOM3:REF<1-4>:HORIZONTAL:POSITION  | 0.0000 |
| ZOOM:ZOOM3:REF<1-4>:HORIZONTAL:SCALE     | 5      |
| ZOOM:ZOOM3:REF<1-4>:VERTICAL:POSITION    | 0.0000 |
| ZOOM:ZOOM3:REF<1-4>:VERTICAL:SCALE       | 1.0000 |

|  |        |
|--|--------|
| ZOOM:ZOOM3:SCROLLLOCK                    | 1      |
| ZOOM:ZOOM3:STATE                         | 0      |
| ZOOM:ZOOM4:CH<1-4>:HORIZONTAL:POSITION   | 0.0000 |
| ZOOM:ZOOM4:CH<1-4>:HORIZONTAL:SCALE      | 5      |
| ZOOM:ZOOM4:CH<1-4>:VERTICAL:POSITION     | 0.0000 |
| ZOOM:ZOOM4:CH<1-4>:VERTICAL:SCALE        | 1.0000 |
| ZOOM:ZOOM4:MATH<1-4>:HORIZONTAL:POSITION | 0.0000 |
| ZOOM:ZOOM4:MATH<1-4>:HORIZONTAL:SCALE    | 5      |
| ZOOM:ZOOM4:MATH<1-4>:VERTICAL:POSITION   | 0.0000 |
| ZOOM:ZOOM4:MATH<1-4>:VERTICAL:SCALE      | 1.0000 |
| ZOOM:ZOOM4:REF<1-4>:HORIZONTAL:POSITION  | 0.0000 |
| ZOOM:ZOOM4:REF<1-4>:HORIZONTAL:SCALE     | 5      |
| ZOOM:ZOOM4:REF<1-4>:VERTICAL:POSITION    | 0.0000 |
| ZOOM:ZOOM4:REF<1-4>:VERTICAL:SCALE       | 1.0000 |
| ZOOM:ZOOM4:SCROLLLOCK                    | 1      |
| ZOOM:ZOOM4:STATE                         | 0      |

---

## Appendix D: GPIB Interface Specifications

### GPIB Interface Specifications

#### Overview

This topic describes details of the GPIB remote interface of the instrument. Normally, you will not need this information to use the instrument, but this information may be useful if you are connecting to controllers with unusual configurations.

#### GPIB Functions

The following table lists the GPIB interface functions and electrical function subsets supported by this instrument and a brief description of each function.

#### Interface Messages

The following table shows the standard interface messages that the instrument supports.





---

# Glossary

## **ASCII**

Acronym for the American Standard Code for Information Interchange.  
Controllers transmit commands to the instrument using ASCII character encoding.

## **Address**



# Index

## A

ACQuire:MAXSamplerate?, 2-51  
ACQuire?, 2-51  
ACQuire:MODE, 2-52  
ACQuire:NUMAcq?, 2-53  
ACQuire:NUMAVg, 2-53  
ACQuire:NUMEnv, 2-54  
ACQuire:STATE, 2-54  
ACQuire:STOPAfter, 2-55  
Acquisition commands  
    ACQuire:MAXSamplerate?, 2-51  
    ACQuire?, 2-51  
    ACQuire:MODE, 2-52  
    ACQuire:NUMAcq?, 2-53  
    ACQuire:NUMAVg, 2-54  
    ACQuire:NUMEnv, 2-54  
    ACQuire:STATE, 2-55  
    ACQuire:STOPAfter, 2-55  
    FASTAcq?, 2-130  
    FASTAcq:STATE, 2-131  
Alias commands  
    ALias, 2-56  
    ALias[:STATE], 2-59  
    ALias:CATalog?, 2-57  
    ALias:DEFine, 2-57  
    ALias:DELEte, 2-58  
    ALias:DELEte[:NAME], 2-58  
    ALias:DELEte:ALL, 2-58  
ALias, 2-56  
ALias[:STATE], 2-59  
ALias:CATalog?, 2-56  
ALias:DEFine, 2-57  
ALias:DELEte, 2-58  
ALias:DELEte[:NAME], 2-58  
ALias:DELEte:ALL, 2-58  
ALLEv?, 2-59  
AUTOSet, 2-60  
AUXout?, 2-60  
AUXout:EDGE, 2-61  
AUXout:SOUrce, 2-61

## B

BELI, 2-62

BUS:B<x>:CAN:BITRate, 2-62  
BUS:B<x>:CAN:PRObe, 2-63  
BUS:B<x>:CAN:SAMPLEpoint, 2-63  
BUS:B<x>:CAN:SOUrce, 2-64  
BUS:B<x>:DISPLAY:FORMAt, 2-64  
BUS:B<x>:DISPLAY:TYPe, 2-65  
BUS:B<x>:I2C{:CLOCK|:SCLK}:SOUrce, 2-65  
BUS:B<x>:I2C{:DATA|:SDATA}:SOUrce, 2-65  
BUS:B<x>:POSition, 2-66  
BUS:B<x>:SPI:DATA{:IN|:MISO}:POLARity, 2-67  
BUS:B<x>:SPI:DATA{:IN|:MISO}:SOUrce, 2-67  
BUS:B<x>:SPI:DATA{:OUT|:MOSI}:  
    POLARity, 2-68  
BUS:B<x>:SPI:DATA{:OUT|:MOSI}:SOUrce, 2-68  
BUS:B<x>:SPI{:CLOCK|:SCLK}:POLARity, 2-66  
BUS:B<x>:SPI{:CLOCK|:SCLK}:SOUrce, 2-66  
BUS:B<x>:SPI{:SElect|:SS}:POLARity, 2-68  
BUS:B<x>:SPI{:SElect|:SS}:SOUrce, 2-69  
BUS:B<x>:STATE, 2-69  
BUS:B<x>:TYPE, 2-70  
BUS:THReshold:CH<x>, 2-70  
BUS?, 2-62  
BUSY?, 2-70

## C

CAL?, 2-71  
\*CAL?, 2-72  
CALibrate:FActory:NOTify:DUE?, 2-73  
CALibrate:FActory:NOTify:HOuRs, 2-73  
CALibrate:FActory:NOTify:YEaRs, 2-74  
CALibrate:FActory:STATus?, 2-74  
CALibrate:FActory:STEPSTatus?, 2-74  
CALibrate:FActory:STEPSTIMulus?, 2-75  
CALibrate:INTERNaL:STARt, 2-76  
CALibrate:INTERNaL:STATus?, 2-76  
CALibrate:RESuLts:FActory:FrequenCy?, 2-78  
CALibrate:RESuLts:FActory:TriggeR?, 2-78  
CALibrate:RESuLts:FActory:VoltagE?, 2-79  
CALibrate:RESuLts:FActory?, 2-78  
CALibrate:RESuLts:SPC?, 2-79  
CALibrate:TEMPerature?, 2-79  
CALibrate:FActory, 2-72  
CALibrate:INTERNaL, 2-75

- CALibrate:REsults?, 2-77
- Calibration and Diagnostic commands
  - CAL?, 2-71
  - \*CAL?, 2-72
  - CALibrate:FAcTory:NOTIfy:DUE?, 2-73
  - CALibrate:FAcTory:NOTIfy:HOuRs, 2-73
  - CALibrate:FAcTory:NOTIfy:YEaRs, 2-74
  - CALibrate:FAcTory:STATus?, 2-74
  - CALibrate:FAcTory:STEPSTatus?, 2-74
  - CALibrate:FAcTory:STEPSTIMulus?, 2-75
  - CALibrate:INTERNaL:STARt, 2-76
  - CALibrate:INTERNaL:STATus?, 2-76
  - CALibrate:REsults:FAcTory:FRequency?, 2-78
  - CALibrate:REsults:FAcTory:TRigger?, 2-78
  - CALibrate:REsults:FAcTory:VoltagE?, 2-79
  - CALibrate:REsults:FAcTory?, 2-78
  - CALibrate:REsults:SPC?, 2-79
  - CALibrate:TEMPerature?, 2-79
  - CALibrate:FAcTory, 2-72
  - CALibrate:INTERNaL, 2-75
  - CALibrate:REsults?, 2-77
  - DIAG:LOOP:OPTion:NTIMes, 2-109
  - DIAG:SElect:<function>, 2-111
  - DIAG:LOOP:OPTion, 2-108
  - DIAG:LOOP:STOP, 2-109
  - DIAG:MODE, 2-109
  - DIAG:RESUlt:FLAg?, 2-110
  - DIAG:RESUlt:LOG?, 2-110
  - DIAG:SElect:ALL, 2-111
  - DIAG:STATE, 2-112
- CH<x>:BANdwidth, 2-80
- CH<x>:COUPling, 2-81
- CH<x>:DESKew, 2-82
- CH<x>:INVert, 2-82
- CH<x>:OFFSet, 2-83
- CH<x>:POSition, 2-84
- CH<x>:PRObe:AUTOZero, 2-85
- CH<x>:PRObe:DEGAUss, 2-85
- CH<x>:PRObe:DEGAUss:STATE?, 2-86
- CH<x>:PRObe:FORCERange, 2-86
- CH<x>:PRObe:GAIN, 2-86
- CH<x>:PRObe:ID:SERnumber?, 2-87
- CH<x>:PRObe:ID:TYPE?, 2-87
- CH<x>:PRObe:ID?, 2-87
- CH<x>:PRObe:RESistance?, 2-88
- CH<x>:PRObe:SIGnal, 2-88
- CH<x>:PRObe:UNIts?, 2-88
- CH<x>:PRObe?, 2-85
- CH<x>:SCAle, 2-89
- CH<x>:TERmination, 2-90
- CH<x>:YUNits, 2-90
- CH<x>?, 2-80
- CLEARMenu, 2-90
- \*CLS, 2-91
- Command Groups, 2-11
- Cursor commands
  - CURSor:HBArS:POSITION<x>, 2-93
  - CURSor:VBArS:HPOS<x>?, 2-96
  - CURSor:VBArS:POSITION<x>, 2-97
  - CURSor:VBArS:VDELtA?, 2-99
  - CURSor?, 2-91
  - CURSor:FUNCTion, 2-92
  - CURSor:HBArS?, 2-93
  - CURSor:HBArS:DELtA?, 2-93
  - CURSor:HBArS:UNIts, 2-94
  - CURSor:HBArS:USE, 2-94
  - CURSor:MODE, 2-95
  - CURSor:VBArS?, 2-95
  - CURSor:VBArS:DELtA?, 2-96
  - CURSor:VBArS:UNIts, 2-97
  - CURSor:VBArS:USE, 2-98
- CURSor:HBArS:POSITION<x>, 2-93
- CURSor:VBArS:HPOS<x>?, 2-96
- CURSor:VBArS:POSITION<x>, 2-97
- CURSor:VBArS:VDELtA?, 2-99
- CURSor?, 2-91
- CURSor:FUNCTion, 2-92
- CURSor:HBArS?, 2-93
- CURSor:HBArS:DELtA?, 2-93
- CURSor:HBArS:UNIts, 2-94
- CURSor:HBArS:USE, 2-94
- CURSor:MODE, 2-95
- CURSor:VBArS?, 2-95
- CURSor:VBArS:DELtA?, 2-96
- CURSor:VBArS:UNIts, 2-97
- CURSor:VBArS:USE, 2-98
- CURVe, 2-99

## D

- DATA, 2-100
- DATA:DESTination, 2-101
- DATA:ENCdg, 2-102
- DATA:SOUrce, 2-104
- DATA:STARt, 2-105

- DATA:STOP, 2-105
- DATE, 2-106
- \*DDT, 2-107
- DESE, 2-107
- DIAG:LOOP:OPTION:NTIMes, 2-109
- DIAG:SElect:<function>, 2-111
- DIAG:LOOP:OPTION, 2-108
- DIAG:LOOP:STOP, 2-109
- DIAG:MODE, 2-109
- DIAG:RESUlt:FLAG?, 2-110
- DIAG:RESUlt:LOG?, 2-110
- DIAG:SElect:ALL, 2-111
- DIAG:STATE, 2-111
- Display commands
  - DISplay:INTENSITy:BACKLight, 2-115
  - DISplay:INTENSITy:GRAticule, 2-115
  - DISplay:INTENSITy:WAVEform, 2-115
  - DISplay:PICTure:AUTOContrast, 2-117
  - DISplay:PICTure:BRiGhtness, 2-117
  - DISplay:PICTure:CONTRAst, 2-118
  - DISplay:PICTure:STATE, 2-118
  - DISplay:STYle:DOTsonly, 2-119
  - DISplay:VECTorscope:STATE, 2-119
  - DISplay:VECTorscope:STsetting, 2-120
  - DISplay:VECTorscope:TYPe, 2-120
  - DISplay:VECTorscope:TYsetting, 2-120
  - DISplay?, 2-112
  - DISplay:CLOCK, 2-113
  - DISplay:FORMAt, 2-113
  - DISplay:GRAticule, 2-114
  - DISplay:INTENSITy?, 2-114
  - DISplay:PERStistence, 2-116
  - DISplay:XY:MODE, 2-121
  - DISplay:XY:YCHannel, 2-122
  - DISplay:XY:YREF, 2-122
  - MESSAge:BOX, 2-199
  - MESSAge:CLEAR, 2-200
  - MESSAge:SHOW, 2-200
  - MESSAge:STATE, 2-204
- DISplay:INTENSITy:BACKLight, 2-115
- DISplay:INTENSITy:GRAticule, 2-115
- DISplay:INTENSITy:WAVEform, 2-115
- DISplay:PICTure:AUTOContrast, 2-116
- DISplay:PICTure:BRiGhtness, 2-117
- DISplay:PICTure:CONTRAst, 2-118
- DISplay:PICTure:STATE, 2-118
- DISplay:STYle:DOTsonly, 2-119
- DISplay:VECTorscope:STATE, 2-119
- DISplay:VECTorscope:STsetting, 2-119
- DISplay:VECTorscope:TYPe, 2-120
- DISplay:VECTorscope:TYsetting, 2-120
- DISplay?, 2-112
- DISplay:CLOCK, 2-113
- DISplay:FORMAt, 2-113
- DISplay:GRAticule, 2-114
- DISplay:INTENSITy?, 2-114
- DISplay:PERStistence, 2-116
- DISplay:XY:MODE, 2-121
- DISplay:XY:YCHannel, 2-122
- DISplay:XY:YREF, 2-122
- ETHERnet commands
  - ETHERnet:DNS:IPADDDress, 2-125
  - ETHERnet:ENET:ADDress?, 2-125
  - ETHERnet:GATEWay:IPADDDress, 2-126
  - ETHERnet:DHCPbootp, 2-124
  - ETHERnet:DOMAINname, 2-125
  - ETHERnet:HTTTPort, 2-126
  - ETHERnet:IPADDDress, 2-126
  - ETHERnet:NAME, 2-127
  - ETHERnet:PASSWord, 2-127
  - ETHERnet:SUBNETMask, 2-128
- ETHERnet:DNS:IPADDDress, 2-125
- ETHERnet:ENET:ADDress?, 2-125
- ETHERnet:GATEWay:IPADDDress, 2-126
- ETHERnet:DHCPbootp, 2-124
- ETHERnet:DOMAINname, 2-125
- ETHERnet:HTTTPort, 2-126
- ETHERnet:IPADDDress, 2-126
- ETHERnet:NAME, 2-127
- ETHERnet:PASSWord, 2-127
- ETHERnet:SUBNETMask, 2-128
- EVENT?, 2-128
- EVMsg?, 2-128
- EVQty?, 2-129

## F

- FACtory, 2-129
- FASTAcq?, 2-130
- FASTAcq:STATE, 2-131

## File System commands

- FILESystem:FREEspace?, 2-134
- FILESystem?, 2-131
- FILESystem:COPy, 2-132
- FILESystem:CWD, 2-133
- FILESystem:DELEte, 2-133
- FILESystem:DIR?, 2-134
- FILESystem:MKDir, 2-134
- FILESystem:REName, 2-135
- FILESystem:RMDir, 2-135
- FILESystem:FREEspace?, 2-134
- FILESystem?, 2-131
- FILESystem:COPy, 2-132
- FILESystem:CWD, 2-132
- FILESystem:DELEte, 2-133
- FILESystem:DIR?, 2-134
- FILESystem:MKDir, 2-134
- FILESystem:REName, 2-135
- FILESystem:RMDir, 2-135
- FPANEL:PRESS, 2-136
- FPANEL:TURN, 2-137

**G**

- GPiBUsb:ADdRes?, 2-137
- GPiBUsb:HWVersion, 2-138
- GPiBUsb:ID?, 2-138
- GPiBUsb:SETADdRes, 2-138
- GPiBUsb:SETID, 2-138
- GPiBUsb:STATUS, 2-138

**H**

## Hard Copy commands

- HARDCopy, 2-139
- HARDCopy:ACTIVeprinter, 2-139
- HARDCopy:PRINTer:ADD, 2-141
- HARDCopy:PRINTer:DELEte, 2-141
- HARDCopy:PRINTer:LIST?, 2-141
- HARDCopy:PRINTer:REName, 2-142
- HARDCopy:INKSaver, 2-140
- HARDCopy:LAYout, 2-140
- HARDCopy:PREVIEW, 2-140
- HARDCopy, 2-139
- HARDCopy:ACTIVeprinter, 2-139
- HARDCopy:PRINTer:ADD, 2-141
- HARDCopy:PRINTer:DELEte, 2-141
- HARDCopy:PRINTer:LIST?, 2-141

- HARDCopy:PRINTer:REName, 2-142

- HARDCopy:INKSaver, 2-139

- HARDCopy:LAYout, 2-140

- HARDCopy:PREVIEW, 2-140

- HDR, 2-142

- HEADer, 2-143

## Horizontal commands

- HORizontal:[MAIn]:DELay:STATe, 2-145
- HORizontal:ACQLENGTH?, 2-144
- HORizontal:MAIn:SAMPLERate?, 2-147
- HORizontal:MAIn:SECdiv, 2-148
- HORizontal:MAIn:UNIts:STRing?, 2-149
- HORizontal:MAIn:UNIts?, 2-148
- HORizontal:PREViewstate?, 2-149
- HORizontal:RECOrdlength, 2-149
- HORizontal:RESOLution, 2-150
- HORizontal?, 2-144
- HORizontal[:MAIn]:DELay:MODE, 2-145
- HORizontal[:MAIn]:DELay:TIME, 2-146
- HORizontal[:MAIn]:POSition?, 2-147
- HORizontal[:MAIn]:SCALE, 2-147
- HORizontal:MAIn?, 2-145
- HORizontal:ROLL?, 2-150
- HORizontal:[MAIn]:DELay:STATe, 2-145
- HORizontal:ACQLENGTH?, 2-144
- HORizontal:MAIn:SAMPLERate?, 2-147
- HORizontal:MAIn:SECdiv, 2-148
- HORizontal:MAIn:UNIts:STRing?, 2-149
- HORizontal:MAIn:UNIts?, 2-148
- HORizontal:PREViewstate?, 2-149
- HORizontal:RECOrdlength, 2-149
- HORizontal:RESOLution, 2-150
- HORizontal?, 2-144
- HORizontal[:MAIn]:DELay:MODE, 2-145
- HORizontal[:MAIn]:DELay:TIME, 2-146
- HORizontal[:MAIn]:POSition?, 2-146
- HORizontal[:MAIn]:SCALE, 2-147
- HORizontal:MAIn?, 2-144
- HORizontal:ROLL?, 2-150

**I**

- ID?, 2-151

- \*IDN?, 2-151

**J**

- JIS, 2-203

**L**

LANGUage, 2-152  
 LANGUage:INCRement, 2-152  
 LOCK, 2-152  
 \*LRN?, 2-153

**M**

## Mark commands

MARK, 2-154  
 MARK:SElected:FOCUS?, 2-155  
 MARK:SElected:MARKSINCOLUMN?, 2-155  
 MARK:SElected:OWNer?, 2-156  
 MARK:SElected:SOURCE?, 2-156  
 MARK:SElected:START?, 2-156  
 MARK:SElected:STATE?, 2-156  
 MARK:SElected:ZOOM:POSITION?, 2-157  
 MARK:CREATE, 2-154  
 MARK:DELEte, 2-154  
 MARK:FREE?, 2-155  
 MARK:SElected:END?, 2-155  
 MARK:TOTAL?, 2-157  
 MARK, 2-154  
 MARK:SElected:FOCUS?, 2-155  
 MARK:SElected:MARKSINCOLUMN?, 2-155  
 MARK:SElected:OWNer?, 2-156  
 MARK:SElected:SOURCE?, 2-156  
 MARK:SElected:START?, 2-156  
 MARK:SElected:STATE?, 2-156  
 MARK:SElected:ZOOM:POSITION?, 2-157  
 MARK:CREATE, 2-154  
 MARK:DELEte, 2-154  
 MARK:FREE?, 2-155  
 MARK:SElected:END?, 2-155  
 MARK:TOTAL?, 2-157

## Math commands

MATH[1]:HORizontal:POSition, 2-158  
 MATH[1]:HORizontal:SCALE, 2-159  
 MATH[1]:HORizontal:UNITs, 2-159  
 MATH[1]:SPECTral:MAG, 2-160  
 MATH[1]:SPECTral:WINDow, 2-161  
 MATH[1]:VERTical:POSition, 2-162  
 MATH[1]:VERTical:SCALE, 2-163  
 MATH[1]:VERTical:UNITs, 2-164  
 MATH[1]:DEFIne, 2-157  
 MATH[1]:TYPE, 2-161  
 MATHVAR:VAR<x>, 2-165  
 MATHVAR?, 2-164

MATH[1]:HORizontal:POSition, 2-158  
 MATH[1]:HORizontal:SCALE, 2-159  
 MATH[1]:HORizontal:UNITs, 2-159  
 MATH[1]:SPECTral:MAG, 2-160  
 MATH[1]:SPECTral:WINDow, 2-160  
 MATH[1]:VERTical:POSition, 2-162  
 MATH[1]:VERTical:SCALE, 2-163  
 MATH[1]:VERTical:UNITs, 2-164  
 MATH[1]:DEFIne, 2-157  
 MATH[1]:TYPE, 2-161  
 MATHVAR:VAR<x>, 2-164  
 MATHVAR?, 2-164

## Measurement commands

MEASUrement:CLEARSnapshot, 2-166  
 MEASUrement:IMMed:DElay:DIREction, 2-168  
 MEASUrement:IMMed:DElay:EDGE<x>, 2-168  
 MEASUrement:IMMed:DElay?, 2-167  
 MEASUrement:IMMed:SOURCE, 2-169  
 MEASUrement:IMMed:SOURCE2, 2-170  
 MEASUrement:IMMed:TYPE, 2-170  
 MEASUrement:IMMed:UNITs?, 2-175  
 MEASUrement:IMMed:VALue?, 2-175  
 MEASUrement:INDICators:HORZ<x>?, 2-176  
 MEASUrement:INDICators:NUMHORZ?, 2-176  
 MEASUrement:INDICators:NUMVERT?, 2-177  
 MEASUrement:INDICators:STATE, 2-177  
 MEASUrement:INDICators:VERT<x>?, 2-178  
 MEASUrement:INDICators?, 2-176  
 MEASUrement:MEAS<x>:COUNT?, 2-179  
 MEASUrement:MEAS<x>:DElay:  
   DIREction, 2-179  
 MEASUrement:MEAS<x>:DElay:  
   EDGE<x>, 2-180  
 MEASUrement:MEAS<x>:DElay?, 2-179  
 MEASUrement:MEAS<x>:MAXimum?, 2-181  
 MEASUrement:MEAS<x>:MEAN?, 2-181  
 MEASUrement:MEAS<x>:MINImum?, 2-181  
 MEASUrement:MEAS<x>:SOURCE[1], 2-182  
 MEASUrement:MEAS<x>:SOURCE2, 2-183  
 MEASUrement:MEAS<x>:STATE, 2-183  
 MEASUrement:MEAS<x>:STDdev?, 2-184  
 MEASUrement:MEAS<x>:TYPE, 2-184  
 MEASUrement:MEAS<x>:UNITs?, 2-189  
 MEASUrement:MEAS<x>:VALue?, 2-189  
 MEASUrement:MEAS<x>?, 2-178  
 MEASUrement:REFLevel:ABSolute:  
   HIGH, 2-191

- MEASUrement:REFLevel:ABSolute:LOW, 2-192
- MEASUrement:REFLevel:ABSolute:MID, 2-193
- MEASUrement:REFLevel:ABSolute:  
MID2, 2-193
- MEASUrement:REFLevel:METHod, 2-194
- MEASUrement:REFLevel:PERCent:  
HIGH, 2-195
- MEASUrement:REFLevel:PERCent:LOW, 2-196
- MEASUrement:REFLevel:PERCent:MID, 2-196
- MEASUrement:REFLevel:PERCent:MID2, 2-197
- MEASUrement:REFLevel?, 2-191
- MEASUrement:SNAPShot, 2-198
- MEASUrement:STATIstics:MODE, 2-198
- MEASUrement:STATIstics:WEIghting, 2-199
- MEASUrement?, 2-165
- MEASUrement:GATing, 2-166
- MEASUrement:IMMed?, 2-167
- MEASUrement:METHod, 2-190
- MEASUrement:CLEARSNapshot, 2-166
- MEASUrement:IMMed:DELAy:DIREction, 2-167
- MEASUrement:IMMed:DELAy:EDGE<x>, 2-168
- MEASUrement:IMMed:DELAy?, 2-167
- MEASUrement:IMMed:SOURCE, 2-169
- MEASUrement:IMMed:SOURCE2, 2-169
- MEASUrement:IMMed:TYPe, 2-170
- MEASUrement:IMMed:UNIts?, 2-175
- MEASUrement:IMMed:VALue?, 2-175
- MEASUrement:INDICators:HORIZ<x>?, 2-176
- MEASUrement:INDICators:NUMHORZ?, 2-176
- MEASUrement:INDICators:NUMVERT?, 2-177
- MEASUrement:INDICators:STATE, 2-177
- MEASUrement:INDICators:VERT<x>?, 2-178
- MEASUrement:INDICators?, 2-176
- MEASUrement:MEAS<x>:COUNT?, 2-179
- MEASUrement:MEAS<x>:DELAy:DIREction, 2-179
- MEASUrement:MEAS<x>:DELAy:EDGE<x>, 2-180
- MEASUrement:MEAS<x>:DELAy?, 2-179
- MEASUrement:MEAS<x>:MAXimum?, 2-181
- MEASUrement:MEAS<x>:MEAN?, 2-181
- MEASUrement:MEAS<x>:MINImum?, 2-181
- MEASUrement:MEAS<x>:SOURCE[1], 2-182
- MEASUrement:MEAS<x>:SOURCE2, 2-182
- MEASUrement:MEAS<x>:STATE, 2-183
- MEASUrement:MEAS<x>:STDdev?, 2-184
- MEASUrement:MEAS<x>:TYPe, 2-184
- MEASUrement:MEAS<x>:UNIts?, 2-189
- MEASUrement:MEAS<x>:VALue?, 2-189
- MEASUrement:MEAS<x>?, 2-178
- MEASUrement:REFLevel:ABSolute:HIGH, 2-191
- MEASUrement:REFLevel:ABSolute:LOW, 2-192
- MEASUrement:REFLevel:ABSolute:MID, 2-192
- MEASUrement:REFLevel:ABSolute:MID2, 2-193
- MEASUrement:REFLevel:METHod, 2-194
- MEASUrement:REFLevel:PERCent:HIGH, 2-195
- MEASUrement:REFLevel:PERCent:LOW, 2-195
- MEASUrement:REFLevel:PERCent:MID, 2-196
- MEASUrement:REFLevel:PERCent:MID2, 2-197
- MEASUrement:REFLevel?, 2-191
- MEASUrement:SNAPShot, 2-198
- MEASUrement:STATIstics:MODE, 2-198
- MEASUrement:STATIstics:WEIghting, 2-199
- MEASUrement?, 2-165
- MEASUrement:GATing, 2-166
- MEASUrement:IMMed?, 2-167
- MEASUrement:METHod, 2-190
- MESSAge:BOX, 2-199
- MESSAge:CLEAR, 2-200
- MESSAge:SHOW, 2-200
- MESSAge:STATE, 2-204



## Miscellaneous commands

AUTOSet, 2-60  
 AUXout?, 2-60  
 AUXout:EDGE, 2-61  
 AUXout:SOUrce, 2-61  
 BELI, 2-62  
 CLEARMenu, 2-91  
 DATE, 2-106  
 \*DDT, 2-107  
 FPANEL:PRESS, 2-136  
 FPANEL:TURN, 2-137  
 GPIBUsb:ADDress?, 2-137  
 GPIBUsb:HWVersion, 2-138  
 GPIBUsb:ID?, 2-138  
 GPIBUsb:SETADDress, 2-138  
 GPIBUsb:SETID, 2-138  
 GPIBUsb:STATUS, 2-138  
 HDR, 2-142  
 HEADer, 2-143  
 ID?, 2-151  
 \*IDN?, 2-151  
 LANGuage, 2-152  
 LANGuage:INCRement, 2-152  
 LOCK, 2-152  
 \*LRN?, 2-153  
 NEWpass, 2-204  
 REM, 2-212  
 SET?, 2-250  
 STArtuptime?, 2-252  
 TEKSecure, 2-252  
 TIME, 2-253  
 TOTAluptime?, 2-253  
 \*TRG, 2-254  
 \*TST?, 2-321  
 UNLock, 2-322  
 VERBoSe, 2-322

**N**

NEWpass, 2-204

**O**

\*OPC, 2-205  
 \*OPT?, 2-206

**P**

\*PSC, 2-206

\*PUD, 2-206

**R**

\*RCL, 2-207  
 RECALL:SETUp, 2-208  
 RECALL:WAVEform, 2-208  
 REF<x>:DATE?, 2-209  
 REF<x>:HORizontal:POSition, 2-209  
 REF<x>:HORizontal:SCALE, 2-210  
 REF<x>:TIME?, 2-210  
 REF<x>:VERTical:POSition, 2-210  
 REF<x>:VERTical:SCALE, 2-211  
 REM, 2-212  
 \*RST, 2-213

**S**

\*SAV, 2-213

## Save and Recall commands

FACtory, 2-130  
 \*RCL, 2-207  
 RECALL:SETUp, 2-208  
 RECALL:WAVEform, 2-208  
 \*SAV, 2-214  
 SAVe:IMAGe:FILEFormat, 2-214  
 SAVe:WAVEform:FILEFormat, 2-217  
 SAVe:IMAGe, 2-214  
 SAVe:SETUp, 2-215  
 SAVe:WAVEform, 2-216  
 SAVe:IMAGe:FILEFormat, 2-214  
 SAVe:WAVEform:FILEFormat, 2-216  
 SAVe:IMAGe, 2-214  
 SAVe:SETUp, 2-215  
 SAVe:WAVEform, 2-216

## Search commands

SEARCH:SEARCH<x>:COPy, 2-218  
 SEARCH:SEARCH<x>:STATE, 2-218  
 SEARCH:SEARCH<x>:TOTAL?, 2-218  
 SEARCH:SEARCH<x>:TRIGger:A:BUS, 2-219  
 SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:  
     CAN:CONDition, 2-219  
 SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:  
     CAN:DATA:DIRection, 2-220  
 SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:  
     CAN:DATA:QUALifier, 2-220  
 SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:  
     CAN:DATA:SIZE, 2-221

SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:  
CAN:DATa:VALue, 2-221  
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:  
CAN:FRAMeType, 2-222  
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:  
CAN{:IDentifier|:ADDRess}:MODE, 2-222  
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:  
CAN{:IDentifier|:ADDRess}:VALue, 2-223  
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:  
I2C:ADDRess:MODE, 2-223  
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:  
I2C:ADDRess:TYPe, 2-223  
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:  
I2C:ADDRess:VALue, 2-224  
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:  
I2C:CONDition, 2-224  
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:  
I2C:DATa:DIRection, 2-225  
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:  
I2C:DATa:SIZE, 2-225  
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:  
I2C:DATa:VALue, 2-226  
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:  
SPI:CONDition, 2-226  
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:  
SPI:DATa:SIZE, 2-227  
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:  
SPI:DATa{:MISO|:IN}:VALue, 2-227  
SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:  
SPI:DATa{:MOSI|:OUT}:VALue, 2-227  
SEARCH:SEARCH<x>:TRIGger:A:BUS:  
SOUrce, 2-228  
SEARCH:SEARCH<x>:TRIGger:A:EDGE:  
SLOpe, 2-228  
SEARCH:SEARCH<x>:TRIGger:A:EDGE:  
SOUrce, 2-228  
SEARCH:SEARCH<x>:TRIGger:A:  
LEVel, 2-229  
SEARCH:SEARCH<x>:TRIGger:A:LEVel:  
CH<x>, 2-229  
SEARCH:SEARCH<x>:TRIGger:A:LEVel:  
MATH, 2-229  
SEARCH:SEARCH<x>:TRIGger:A:LEVel:  
REF<x>, 2-230  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:  
FUNction, 2-230  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:  
INPut:CH<x>, 2-231  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:  
INPut:CLOCK:EDGE, 2-231  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:  
INPut:CLOCK:SOUrce, 2-231  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:  
INPut:MATH, 2-232  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:  
INPut:REF<x>, 2-232  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:  
PATtern:INPut:CH<x>, 2-233  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:  
PATtern:INPut:MATH, 2-233  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:  
PATtern:INPut:REF<x>, 2-233  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:  
PATtern:WHEn, 2-234  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:  
PATtern:WHEn:LESSLimit, 2-235  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:  
PATtern:WHEn:MORELimit, 2-235  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:  
THReshold:CH<x>, 2-235  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:  
THReshold:MATH, 2-236  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:  
THReshold:REF<x>, 2-236  
SEARCH:SEARCH<x>:TRIGger:A:  
LOWerthreshold:CH<x>, 2-236  
SEARCH:SEARCH<x>:TRIGger:A:  
LOWerthreshold:MATH, 2-237  
SEARCH:SEARCH<x>:TRIGger:A:  
LOWerthreshold:REF<x>, 2-237  
SEARCH:SEARCH<x>:TRIGger:A:  
PULSEWidth:POLarity, 2-237  
SEARCH:SEARCH<x>:TRIGger:A:  
PULSEWidth:SOUrce, 2-238  
SEARCH:SEARCH<x>:TRIGger:A:  
PULSEWidth:WHEn, 2-238  
SEARCH:SEARCH<x>:TRIGger:A:  
PULSEWidth:WIDth, 2-239  
SEARCH:SEARCH<x>:TRIGger:A:RUNT:  
POLarity, 2-239  
SEARCH:SEARCH<x>:TRIGger:A:RUNT:  
SOUrce, 2-240

- SEARCH:SEARCH<x>:TRIGger:A:RUNT:  
WHEN, 2-240
- SEARCH:SEARCH<x>:TRIGger:A:RUNT:  
WIDTH, 2-241
- SEARCH:SEARCH<x>:TRIGger:A:SETHold:  
CLOCK:EDGE, 2-241
- SEARCH:SEARCH<x>:TRIGger:A:SETHold:  
CLOCK:SOUrce, 2-241
- SEARCH:SEARCH<x>:TRIGger:A:SETHold:  
CLOCK:THReshold, 2-242
- SEARCH:SEARCH<x>:TRIGger:A:SETHold:  
DATa:SOUrce, 2-242
- SEARCH:SEARCH<x>:TRIGger:A:SETHold:  
DATa:THReshold, 2-243
- SEARCH:SEARCH<x>:TRIGger:A:SETHold:  
HOLDTime, 2-243
- SEARCH:SEARCH<x>:TRIGger:A:SETHold:  
SETTime, 2-243
- SEARCH:SEARCH<x>:TRIGger:A:TYPE, 2-245
- SEARCH:SEARCH<x>:TRIGger:A:  
UPPerthreshold:CH<x>, 2-246
- SEARCH:SEARCH<x>:TRIGger:A:  
UPPerthreshold:MATH, 2-246
- SEARCH:SEARCH<x>:TRIGger:A:  
UPPerthreshold:REF<x>, 2-247
- SEARCH:SEARCH<x>:TRIGger:A {:  
TRANSition|:RISEFall}:DELTAtime, 2-244
- SEARCH:SEARCH<x>:TRIGger:A {:  
TRANSition|:RISEFall}:POLarity, 2-244
- SEARCH:SEARCH<x>:TRIGger:A {:  
TRANSition|:RISEFall}:SOUrce, 2-244
- SEARCH:SEARCH<x>:TRIGger:A {:  
TRANSition|:RISEFall}:WHEN, 2-245
- SEARCH:SEARCH<x>:COPy, 2-218
- SEARCH:SEARCH<x>:STATE, 2-218
- SEARCH:SEARCH<x>:TOTAL?, 2-218
- SEARCH:SEARCH<x>:TRIGger:A:BUS, 2-219
- SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:  
CAN:CONDition, 2-219
- SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:  
CAN:DATa:DIRection, 2-220
- SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:  
CAN:DATa:QUALifier, 2-220
- SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:  
CAN:DATa:SIZE, 2-221
- SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:  
CAN:DATa:VALue, 2-221
- SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:  
CAN:FRAMEtype, 2-221
- SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:  
CAN { :IDentifier|:ADDReSS } :MODE, 2-222
- SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:  
CAN { :IDentifier|:ADDReSS } :VALue, 2-222
- SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:  
ADDReSS:MODE, 2-223
- SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:  
ADDReSS:TYPE, 2-223
- SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:  
ADDReSS:VALue, 2-224
- SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:  
CONDition, 2-224
- SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:  
DATa:DIRection, 2-225
- SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:  
DATa:SIZE, 2-225
- SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:I2C:  
DATa:VALue, 2-226
- SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:SPI:  
CONDition, 2-226
- SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:SPI:  
DATa:SIZE, 2-227
- SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:SPI:  
DATa { :MISO|:IN } :VALue, 2-226
- SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:SPI:  
DATa { :MOSI|:OUT } :VALue, 2-227
- SEARCH:SEARCH<x>:TRIGger:A:BUS:  
SOUrce, 2-228
- SEARCH:SEARCH<x>:TRIGger:A:EDGE:  
SLOPe, 2-228
- SEARCH:SEARCH<x>:TRIGger:A:EDGE:  
SOUrce, 2-228
- SEARCH:SEARCH<x>:TRIGger:A:LEVel, 2-229
- SEARCH:SEARCH<x>:TRIGger:A:LEVel:  
CH<x>, 2-229
- SEARCH:SEARCH<x>:TRIGger:A:LEVel:  
MATH, 2-229
- SEARCH:SEARCH<x>:TRIGger:A:LEVel:  
REF<x>, 2-230
- SEARCH:SEARCH<x>:TRIGger:A:LOGIc:  
FUNction, 2-230
- SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPut:  
CH<x>, 2-231
- SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPut:  
CLOCK:EDGE, 2-231

SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPut:  
CLOCK:SOUrce, 2-231  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPut:  
MATH, 2-232  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPut:  
REF<x>, 2-232  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PATtern:  
INPut:CH<x>, 2-233  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PATtern:  
INPut:MATH, 2-233  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PATtern:  
INPut:REF<x>, 2-233  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PATtern:  
WHEn, 2-234  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PATtern:  
WHEn:LESSLimit, 2-235  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PATtern:  
WHEn:MORELimit, 2-235  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:  
THReshold:CH<x>, 2-235  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:  
THReshold:MATH, 2-236  
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:  
THReshold:REF<x>, 2-236  
SEARCH:SEARCH<x>:TRIGger:A:  
LOWerthreshold:CH<x>, 2-236  
SEARCH:SEARCH<x>:TRIGger:A:  
LOWerthreshold:MATH, 2-237  
SEARCH:SEARCH<x>:TRIGger:A:  
LOWerthreshold:REF<x>, 2-237  
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:  
POLarity, 2-237  
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:  
SOUrce, 2-238  
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:  
WHEn, 2-238  
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:  
WIDth, 2-239  
SEARCH:SEARCH<x>:TRIGger:A:RUNT:  
POLarity, 2-239  
SEARCH:SEARCH<x>:TRIGger:A:RUNT:  
SOUrce, 2-240  
SEARCH:SEARCH<x>:TRIGger:A:RUNT:  
WHEn, 2-240  
SEARCH:SEARCH<x>:TRIGger:A:RUNT:  
WIDth, 2-241  
SEARCH:SEARCH<x>:TRIGger:A:SETHold:  
CLOCK:EDGE, 2-241  
SEARCH:SEARCH<x>:TRIGger:A:SETHold:  
CLOCK:SOUrce, 2-241  
SEARCH:SEARCH<x>:TRIGger:A:SETHold:  
CLOCK:THReshold, 2-242  
SEARCH:SEARCH<x>:TRIGger:A:SETHold:DATA:  
SOUrce, 2-242  
SEARCH:SEARCH<x>:TRIGger:A:SETHold:DATA:  
THReshold, 2-243  
SEARCH:SEARCH<x>:TRIGger:A:SETHold:  
HOLDTime, 2-243  
SEARCH:SEARCH<x>:TRIGger:A:SETHold:  
SETTime, 2-243  
SEARCH:SEARCH<x>:TRIGger:A:TYPE, 2-245  
SEARCH:SEARCH<x>:TRIGger:A:UPPerthreshold:  
CH<x>, 2-246  
SEARCH:SEARCH<x>:TRIGger:A:UPPerthreshold:  
MATH, 2-246  
SEARCH:SEARCH<x>:TRIGger:A:UPPerthreshold:  
REF<x>, 2-247  
SEARCH:SEARCH<x>:TRIGger:A{:TRANsition|:  
RISEFall}:DELTatime, 2-244  
SEARCH:SEARCH<x>:TRIGger:A{:TRANsition|:  
RISEFall}:POLarity, 2-244  
SEARCH:SEARCH<x>:TRIGger:A{:TRANsition|:  
RISEFall}:SOUrce, 2-244  
SEARCH:SEARCH<x>:TRIGger:A{:TRANsition|:  
RISEFall}:WHEn, 2-245  
SElect:BUS<x>, 2-247  
SElect:CH<x>, 2-248  
SElect:REF<x>, 2-250  
SElect?, 2-247  
SElect:CONTROL, 2-248  
SElect:MATH[1], 2-249  
SET?, 2-250  
\*SRE, 2-251  
STArtuptime?, 2-252

## Status and Error commands

ALLEv?, 2-59  
 BUSY?, 2-71  
 \*CLS, 2-91  
 DESE, 2-107  
 \*ESE, 2-123  
 \*ESR?, 2-124  
 EVENT?, 2-128  
 EVMsg?, 2-128  
 EVQty?, 2-129  
 \*OPC, 2-205  
 \*OPT?, 2-206  
 \*PSC, 2-206  
 \*PUD, 2-207  
 \*RST, 2-213  
 \*SRE, 2-251  
 \*STB?, 2-252  
 \*WAI, 2-323  
 \*STB?, 2-252

## T

TEKSecure, 2-252  
 TIME, 2-253  
 TOTaluptime?, 2-253  
 \*TRG, 2-253

## Trigger commands

BUS:B<x>:CAN:BITRate, 2-62  
 BUS:B<x>:CAN:PRObe, 2-63  
 BUS:B<x>:CAN:SAMPLEpoint, 2-64  
 BUS:B<x>:CAN:SOUrce, 2-64  
 BUS:B<x>:DISPLAY:FORMAT, 2-64  
 BUS:B<x>:DISPLAY:TYPe, 2-65  
 BUS:B<x>:I2C{:CLOCK|:SCLK}:SOUrce, 2-65  
 BUS:B<x>:I2C{:DATA|:SDATA}:SOUrce, 2-65  
 BUS:B<x>:POSition, 2-66  
 BUS:B<x>:SPI:DATA{:IN|:MISO}:  
   POLARity, 2-67  
 BUS:B<x>:SPI:DATA{:IN|:MISO}:SOUrce, 2-67  
 BUS:B<x>:SPI:DATA{:OUT|:MOSI}:  
   POLARity, 2-68  
 BUS:B<x>:SPI:DATA{:OUT|:MOSI}:  
   SOUrce, 2-68  
 BUS:B<x>:SPI{:CLOCK|:SCLK}:  
   POLARity, 2-66  
 BUS:B<x>:SPI{:CLOCK|:SCLK}:SOUrce, 2-67  
 BUS:B<x>:SPI{:SElect|:SS}:POLARity, 2-68  
 BUS:B<x>:SPI{:SElect|:SS}:SOUrce, 2-69

BUS:B<x>:STATE, 2-69  
 BUS:B<x>:TYPE, 2-70  
 BUS:THReshold:CH<x>, 2-70  
 BUS?, 2-62  
 TRIGger, 2-254  
 TRIGger:A:BUS:B<x>:CAN:CONDition, 2-256  
 TRIGger:A:BUS:B<x>:CAN:DATA:  
   DIRection, 2-257  
 TRIGger:A:BUS:B<x>:CAN:DATA:  
   QUALifier, 2-257  
 TRIGger:A:BUS:B<x>:CAN:DATA:SIZE, 2-258  
 TRIGger:A:BUS:B<x>:CAN:DATA:  
   VALue, 2-258  
 TRIGger:A:BUS:B<x>:CAN:FRAMEtype, 2-259  
 TRIGger:A:BUS:B<x>:CAN{:IDentifier|:  
   ADDRes}:MODE, 2-259  
 TRIGger:A:BUS:B<x>:CAN{:IDentifier|:  
   ADDRes}:VALue, 2-260  
 TRIGger:A:BUS:B<x>:I2C:ADDRes:  
   MODE, 2-260  
 TRIGger:A:BUS:B<x>:I2C:ADDRes:  
   TYPe, 2-260  
 TRIGger:A:BUS:B<x>:I2C:ADDRes:  
   VALue, 2-261  
 TRIGger:A:BUS:B<x>:I2C:CONDition, 2-261  
 TRIGger:A:BUS:B<x>:I2C:DATA:  
   DIRection, 2-262  
 TRIGger:A:BUS:B<x>:I2C:DATA:SIZE, 2-263  
 TRIGger:A:BUS:B<x>:I2C:DATA:VALue, 2-263  
 TRIGger:A:BUS:B<x>:SPI:CONDition, 2-263  
 TRIGger:A:BUS:B<x>:SPI:DATA:SIZE, 2-265  
 TRIGger:A:BUS:B<x>:SPI:DATA{:IN|:MISO}:  
   VALue, 2-264  
 TRIGger:A:BUS:B<x>:SPI:DATA{:OUT|:  
   MOSI}:VALue, 2-264  
 TRIGger:A:BUS:SOUrce, 2-265  
 TRIGger:A:EDGE:COUpling, 2-266  
 TRIGger:A:EDGE:SLOpe, 2-267  
 TRIGger:A:EDGE:SOUrce, 2-267  
 TRIGger:A:HOLDoff:TIME, 2-268  
 TRIGger:A:LEVel:CH<x>, 2-269  
 TRIGger:A:LOGIc:CLASs, 2-270  
 TRIGger:A:LOGIc:FUNCTion, 2-271  
 TRIGger:A:LOGIc:INPut:CH<x>, 2-273  
 TRIGger:A:LOGIc:INPut:CLOCK:EDGE, 2-273  
 TRIGger:A:LOGIc:INPut:CLOCK:SOUrce, 2-274  
 TRIGger:A:LOGIc:INPut?, 2-272

- TRIGger:A:LOGIc:PATtern:INPut:CH2, 2-275
- TRIGger:A:LOGIc:PATtern:INPut:CH4, 2-275
- TRIGger:A:LOGIc:PATtern:WHEn, 2-276
- TRIGger:A:LOGIc:PATtern:WHEn:
  - LESSLimit, 2-277
- TRIGger:A:LOGIc:PATtern:WHEn:
  - MORELimit, 2-278
- TRIGger:A:LOGIc:PATtern?, 2-274
- TRIGger:A:LOGIc:SETHold:CLOCK:
  - EDGE, 2-278
- TRIGger:A:LOGIc:SETHold:CLOCK:
  - SOUrce, 2-279
- TRIGger:A:LOGIc:SETHold:CLOCK:
  - THReshold, 2-279
- TRIGger:A:LOGIc:SETHold:DATA:
  - SOUrce, 2-279
- TRIGger:A:LOGIc:SETHold:DATA:
  - THReshold, 2-279
- TRIGger:A:LOGIc:SETHold:HOLDTime, 2-280
- TRIGger:A:LOGIc:SETHold:SETTime, 2-280
- TRIGger:A:LOGIc:THReshold:CH<x>, 2-281
- TRIGger:A:LOGIc:THReshold?, 2-280
- TRIGger:A:LOWerthreshold:CH<x>, 2-281
- TRIGger:A:PULse:CLAss, 2-283
- TRIGger:A:PULse:RUNT:HIGHLimit, 2-284
- TRIGger:A:PULse:RUNT:LOWLimit, 2-284
- TRIGger:A:PULse:RUNT:POLarity, 2-284
- TRIGger:A:PULse:RUNT:SOUrce, 2-284
- TRIGger:A:PULse:RUNT:THReshold:
  - HIGH, 2-285
- TRIGger:A:PULse:RUNT:THReshold:
  - LOW, 2-285
- TRIGger:A:PULse:RUNT:THReshold?, 2-285
- TRIGger:A:PULse:RUNT:WHEn, 2-285
- TRIGger:A:PULse:RUNT:WIDth, 2-286
- TRIGger:A:PULse:WIDth:HIGHLimit, 2-289
- TRIGger:A:PULse:WIDth:LEVel, 2-289
- TRIGger:A:PULse:WIDth:LOWLimit, 2-290
- TRIGger:A:PULse:WIDth:POLarity, 2-290
- TRIGger:A:PULse:WIDth:SOUrce, 2-290
- TRIGger:A:PULse:WIDth:WHEn, 2-291
- TRIGger:A:PULse:WIDth:WIDth, 2-291
- TRIGger:A:PULse{:TRANsition|:SLEWRate}:
  - DELtetime, 2-286
- TRIGger:A:PULse{:TRANsition|:SLEWRate}:
  - HIGHLimit, 2-287
- TRIGger:A:PULse{:TRANsition|:SLEWRate}:
  - LOWLimit, 2-287
- TRIGger:A:PULse{:TRANsition|:SLEWRate}:
  - POLarity, 2-287
- TRIGger:A:PULse{:TRANsition|:SLEWRate}:
  - SOUrce, 2-287
- TRIGger:A:PULse{:TRANsition|:SLEWRate}:
  - THReshold:HIGH, 2-288
- TRIGger:A:PULse{:TRANsition|:SLEWRate}:
  - THReshold:LOW, 2-288
- TRIGger:A:PULse{:TRANsition|:SLEWRate}:
  - THReshold?, 2-288
- TRIGger:A:PULse{:TRANsition|:SLEWRate}:
  - WHEn, 2-289
- TRIGger:A:PULSEWidth:POLarity, 2-291
- TRIGger:A:PULSEWidth:SOUrce, 2-292
- TRIGger:A:PULSEWidth:WHEn, 2-293
- TRIGger:A:PULSEWidth:WIDth, 2-293
- TRIGger:A:PULSEWIDth?, 2-291
- TRIGger:A:RUNT:POLarity, 2-294
- TRIGger:A:RUNT:SOUrce, 2-295
- TRIGger:A:RUNT:WIDth, 2-296
- TRIGger:A:SETHold:CLOCK:EDGE, 2-297
- TRIGger:A:SETHold:CLOCK:SOUrce, 2-298
- TRIGger:A:SETHold:CLOCK:THReshold, 2-298
- TRIGger:A:SETHold:CLOCK?, 2-297
- TRIGger:A:SETHold:DATA:SOUrce, 2-299
- TRIGger:A:SETHold:DATA:THReshold, 2-300
- TRIGger:A:SETHold:DATA?, 2-299
- TRIGger:A:SETHold:HOLDTime, 2-300
- TRIGger:A:SETHold:SETTime, 2-301
- TRIGger:A:UPPerthreshold:CH<x>, 2-305
- TRIGger:A:VIDeo:CUSTom:SCAN, 2-307
- TRIGger:A:VIDeo:CUSTom?, 2-306
- TRIGger:A:VIDeo:CUSTom{:FORMAt|:TYPE}, 2-306
- TRIGger:A:VIDeo:HDtv:FORMAt, 2-307
- TRIGger:A:VIDeo:HOLDoff:FIELD, 2-308
- TRIGger:A:VIDeo:LINE, 2-309
- TRIGger:A:VIDeo:POLarity, 2-310
- TRIGger:A:VIDeo:SOUrce, 2-311
- TRIGger:A:VIDeo:STANDard, 2-311
- TRIGger:A:VIDeo{:SYNC|:FIELD}, 2-312
- TRIGger:A{:TRANsition|:RISEFall}:
  - DELtetime, 2-302
- TRIGger:A{:TRANsition|:RISEFall}:
  - POLarity, 2-302

- TRIGger:A{:TRANSition|:RISEFall}:  
SOURce, 2-303
- TRIGger:A{:TRANSition|:RISEFall}:  
WHEn, 2-303
- TRIGger:A{:TRANSition|:RISEFall}?, 2-301
- TRIGger:B:EDGE:COUPling, 2-314
- TRIGger:B:EDGE:SLOpe, 2-315
- TRIGger:B:EDGE:SOURce, 2-315
- TRIGger:B:EVENTS:COUNt, 2-316
- TRIGger:B:LEVel:CH<x>, 2-318
- TRIGger:B:LOWerthreshold:CH<x>, 2-318
- TRIGger:B:UPPerthreshold:CH<x>, 2-320
- TRIGger:A, 2-254
- TRIGger:A:BUS, 2-256
- TRIGger:A:EDGE?, 2-265
- TRIGger:A:HOLDoff?, 2-268
- TRIGger:A:LEVel, 2-269
- TRIGger:A:LOGIc?, 2-270
- TRIGger:A:MODE, 2-282
- TRIGger:A:PULse?, 2-282
- TRIGger:A:RUNT?, 2-294
- TRIGger:A:RUNT:WHEn, 2-295
- TRIGger:A:SETHold?, 2-296
- TRIGger:A:TYPe, 2-304
- TRIGger:A:VIDeo?, 2-305
- TRIGger:B, 2-312
- TRIGger:B:BY, 2-313
- TRIGger:B:EDGE?, 2-314
- TRIGger:B:EVENTS?, 2-316
- TRIGger:B:LEVel, 2-317
- TRIGger:B:STATE, 2-319
- TRIGger:B:TiMe, 2-319
- TRIGger:B:TYPe, 2-320
- TRIGger:STATE?, 2-321
- TRIGger, 2-254
- TRIGger:A:BUS:B<x>:CAN:CONDition, 2-256
- TRIGger:A:BUS:B<x>:CAN:DATa:DiRection, 2-257
- TRIGger:A:BUS:B<x>:CAN:DATa:QUALifier, 2-257
- TRIGger:A:BUS:B<x>:CAN:DATa:SiZe, 2-258
- TRIGger:A:BUS:B<x>:CAN:DATa:VALue, 2-258
- TRIGger:A:BUS:B<x>:CAN:FRAMeType, 2-258
- TRIGger:A:BUS:B<x>:CAN{:IDentifier|:ADDReSS}:  
MODE, 2-259
- TRIGger:A:BUS:B<x>:CAN{:IDentifier|:ADDReSS}:  
VALue, 2-259
- TRIGger:A:BUS:B<x>:I2C:ADDReSS:MODE, 2-260
- TRIGger:A:BUS:B<x>:I2C:ADDReSS:TYPe, 2-260
- TRIGger:A:BUS:B<x>:I2C:ADDReSS:VALue, 2-261
- TRIGger:A:BUS:B<x>:I2C:CONDition, 2-261
- TRIGger:A:BUS:B<x>:I2C:DATa:DiRection, 2-262
- TRIGger:A:BUS:B<x>:I2C:DATa:SiZe, 2-263
- TRIGger:A:BUS:B<x>:I2C:DATa:VALue, 2-263
- TRIGger:A:BUS:B<x>:SPI:CONDition, 2-263
- TRIGger:A:BUS:B<x>:SPI:DATa:SiZe, 2-265
- TRIGger:A:BUS:B<x>:SPI:DATa{:IN|:MISO}:  
VALue, 2-264
- TRIGger:A:BUS:B<x>:SPI:DATa{:OUT|:MOSI}:  
VALue, 2-264
- TRIGger:A:BUS:SOURce, 2-265
- TRIGger:A:EDGE:COUPling, 2-266
- TRIGger:A:EDGE:SLOpe, 2-267
- TRIGger:A:EDGE:SOURce, 2-267
- TRIGger:A:HOLDoff:TiMe, 2-268
- TRIGger:A:LEVel:CH<x>, 2-269
- TRIGger:A:LOGIc:CLAss, 2-270
- TRIGger:A:LOGIc:FUNCTion, 2-271
- TRIGger:A:LOGIc:INPut:CH<x>, 2-272
- TRIGger:A:LOGIc:INPut:CLOCK:EDGE, 2-273
- TRIGger:A:LOGIc:INPut:CLOCK:SOURce, 2-273
- TRIGger:A:LOGIc:INPut?, 2-272
- TRIGger:A:LOGIc:PATtern:INPut:CH2, 2-274
- TRIGger:A:LOGIc:PATtern:INPut:CH4, 2-275
- TRIGger:A:LOGIc:PATtern:WHEn, 2-276
- TRIGger:A:LOGIc:PATtern:WHEn:  
LESSLimit, 2-277
- TRIGger:A:LOGIc:PATtern:WHEn:  
MORELimit, 2-278
- TRIGger:A:LOGIc:PATtern?, 2-274
- TRIGger:A:LOGIc:SETHold:CLOCK:EDGE, 2-278
- TRIGger:A:LOGIc:SETHold:CLOCK:SOURce, 2-278
- TRIGger:A:LOGIc:SETHold:CLOCK:  
THReShold, 2-279
- TRIGger:A:LOGIc:SETHold:DATa:SOURce, 2-279
- TRIGger:A:LOGIc:SETHold:DATa:  
THReShold, 2-279
- TRIGger:A:LOGIc:SETHold:HOLDTime, 2-280
- TRIGger:A:LOGIc:SETHold:SETTime, 2-280
- TRIGger:A:LOGIc:THReShold:CH<x>, 2-281
- TRIGger:A:LOGIc:THReShold?, 2-280
- TRIGger:A:LOWerthreshold:CH<x>, 2-281
- TRIGger:A:PULse:CLAss, 2-283
- TRIGger:A:PULse:RUNT:HIGHLimit, 2-283
- TRIGger:A:PULse:RUNT:LOWLimit, 2-284
- TRIGger:A:PULse:RUNT:POLarity, 2-284

- TRIGger:A:PULse:RUNT:SOUrce, 2-284
- TRIGger:A:PULse:RUNT:THReshold:HIGH, 2-285
- TRIGger:A:PULse:RUNT:THReshold:LOW, 2-285
- TRIGger:A:PULse:RUNT:THReshold?, 2-285
- TRIGger:A:PULse:RUNT:WHEn, 2-285
- TRIGger:A:PULse:RUNT:WIDth, 2-286
- TRIGger:A:PULse:WIDth:HIGHLimit, 2-289
- TRIGger:A:PULse:WIDth:LEVel, 2-289
- TRIGger:A:PULse:WIDth:LOWLimit, 2-289
- TRIGger:A:PULse:WIDth:POLarity, 2-290
- TRIGger:A:PULse:WIDth:SOUrce, 2-290
- TRIGger:A:PULse:WIDth:WHEn, 2-291
- TRIGger:A:PULse:WIDth:WIDth, 2-291
- TRIGger:A:PULse{:TRANsition|:SLEWRate}:  
DELtAtime, 2-286
- TRIGger:A:PULse{:TRANsition|:SLEWRate}:  
HIGHLimit, 2-287
- TRIGger:A:PULse{:TRANsition|:SLEWRate}:  
LOWLimit, 2-287
- TRIGger:A:PULse{:TRANsition|:SLEWRate}:  
POLarity, 2-287
- TRIGger:A:PULse{:TRANsition|:SLEWRate}:  
SOUrce, 2-287
- TRIGger:A:PULse{:TRANsition|:SLEWRate}:  
THReshold:HIGH, 2-288
- TRIGger:A:PULse{:TRANsition|:SLEWRate}:  
THReshold:LOW, 2-288
- TRIGger:A:PULse{:TRANsition|:SLEWRate}:  
THReshold?, 2-288
- TRIGger:A:PULse{:TRANsition|:SLEWRate}:  
WHEn, 2-289
- TRIGger:A:PULSEWIDth:POLarity, 2-291
- TRIGger:A:PULSEWIDth:SOUrce, 2-292
- TRIGger:A:PULSEWIDth:WHEn, 2-292
- TRIGger:A:PULSEWIDth:WIDth, 2-293
- TRIGger:A:PULSEWIDth?, 2-291
- TRIGger:A:RUNT:POLarity, 2-294
- TRIGger:A:RUNT:SOUrce, 2-295
- TRIGger:A:RUNT:WIDth, 2-296
- TRIGger:A:SETHold:CLOCK:EDGE, 2-297
- TRIGger:A:SETHold:CLOCK:SOUrce, 2-298
- TRIGger:A:SETHold:CLOCK:THReshold, 2-298
- TRIGger:A:SETHold:CLOCK?, 2-297
- TRIGger:A:SETHold:DATA:SOUrce, 2-299
- TRIGger:A:SETHold:DATA:THReshold, 2-300
- TRIGger:A:SETHold:DATA?, 2-299
- TRIGger:A:SETHold:HOLDTime, 2-300
- TRIGger:A:SETHold:SETTime, 2-301
- TRIGger:A:UPPerthreshold:CH<x>, 2-305
- TRIGger:A:VIDeo:CUSTom:SCAN, 2-307
- TRIGger:A:VIDeo:CUSTom?, 2-306
- TRIGger:A:VIDeo:CUSTom{:FORMat|:  
TYPE}, 2-306
- TRIGger:A:VIDeo:HDtv:FORMat, 2-307
- TRIGger:A:VIDeo:HOLDoff:FIELD, 2-308
- TRIGger:A:VIDeo:LINE, 2-309
- TRIGger:A:VIDeo:POLarity, 2-310
- TRIGger:A:VIDeo:SOUrce, 2-310
- TRIGger:A:VIDeo:STANdard, 2-311
- TRIGger:A:VIDeo{:SYNC|:FIELD}, 2-312
- TRIGger:A{:TRANsition|:RISEFall}:  
DELtAtime, 2-302
- TRIGger:A{:TRANsition|:RISEFall}:POLarity, 2-302
- TRIGger:A{:TRANsition|:RISEFall}:SOUrce, 2-303
- TRIGger:A{:TRANsition|:RISEFall}:WHEn, 2-303
- TRIGger:A{:TRANsition|:RISEFall}?, 2-301
- TRIGger:B:EDGE:COUPling, 2-314
- TRIGger:B:EDGE:SLOpe, 2-315
- TRIGger:B:EDGE:SOUrce, 2-315
- TRIGger:B:EVENTS:COUNt, 2-316
- TRIGger:B:LEVel:CH<x>, 2-318
- TRIGger:B:LOWerthreshold:CH<x>, 2-318
- TRIGger:B:UPPerthreshold:CH<x>, 2-320
- TRIGger:A, 2-254
- TRIGger:A:BUS, 2-256
- TRIGger:A:EDGE?, 2-265
- TRIGger:A:HOLDoff?, 2-268
- TRIGger:A:LEVel, 2-269
- TRIGger:A:LOGIc?, 2-270
- TRIGger:A:MODE, 2-282
- TRIGger:A:PULse?, 2-282
- TRIGger:A:RUNT?, 2-294
- TRIGger:A:RUNT:WHEn, 2-295
- TRIGger:A:SETHold?, 2-296
- TRIGger:A:TYPE, 2-304
- TRIGger:A:VIDeo?, 2-305
- TRIGger:B, 2-312
- TRIGger:B:BY, 2-313
- TRIGger:B:EDGE?, 2-314
- TRIGger:B:EVENTS?, 2-316
- TRIGger:B:LEVel, 2-317
- TRIGger:B:STATE, 2-318
- TRIGger:B:TIME, 2-319
- TRIGger:B:TYPE, 2-320



TRIGger:STATE?, 2-320  
\*TST?, 2-321

## U

Unicode, 2-203  
UNLock, 2-321

## V

VERBoSe, 2-322  
Vertical commands  
    CH<x>:BANdwidth, 2-80  
    CH<x>:COUPLing, 2-81  
    CH<x>:DESKew, 2-82  
    CH<x>:INVert, 2-82  
    CH<x>:OFFSet, 2-84  
    CH<x>:POSition, 2-84  
    CH<x>:PRObe:AUTOZero, 2-85  
    CH<x>:PRObe:DEGAUss, 2-85  
    CH<x>:PRObe:DEGAUss:STATE?, 2-86  
    CH<x>:PRObe:FORCERange, 2-86  
    CH<x>:PRObe:GAIN, 2-86  
    CH<x>:PRObe:ID:SERnumber?, 2-87  
    CH<x>:PRObe:ID:TYPE?, 2-88  
    CH<x>:PRObe:ID?, 2-87  
    CH<x>:PRObe:RESistance?, 2-88  
    CH<x>:PRObe:SIGNAL, 2-88  
    CH<x>:PRObe:UNIts?, 2-88  
    CH<x>:PRObe?, 2-85  
    CH<x>:SCALE, 2-89  
    CH<x>:TERmination, 2-90  
    CH<x>:YUNits, 2-90  
    CH<x>?, 2-80  
    REF<x>:DATE?, 2-209  
    REF<x>:HORizontal:POSition, 2-209  
    REF<x>:HORizontal:SCALE, 2-210  
    REF<x>:TIME?, 2-210  
    REF<x>:VERTical:POSition, 2-211  
    REF<x>:VERTical:SCALE, 2-212  
    SElect:BUS<x>, 2-247  
    SElect:CH<x>, 2-248  
    SElect:REF<x>, 2-250  
    SElect?, 2-247  
    SElect:CONTROL, 2-249  
    SElect:MATH[1], 2-249

## W

\*WAI, 2-323  
Waveform Transfer commands  
    CURVe, 2-99  
    DATA, 2-101  
    DATA:DESTination, 2-101  
    DATA:ENCdg, 2-102  
    DATA:SOUrce, 2-104  
    DATA:STARt, 2-105  
    DATA:STOP, 2-106  
    WAVFrm?, 2-323  
    WFMInpre?, 2-324  
    WFMInpre:BIT\_Nr, 2-324  
    WFMInpre:BN\_Fmt, 2-325  
    WFMInpre:BYT\_Nr, 2-325  
    WFMInpre:BYT\_Or, 2-326  
    WFMInpre:ENCdg, 2-326  
    WFMInpre:NR\_Pt, 2-327  
    WFMInpre:PT\_Fmt, 2-328  
    WFMInpre:PT\_Off, 2-329  
    WFMInpre:WFIId, 2-329  
    WFMInpre:XINcr, 2-329  
    WFMInpre:XUNit, 2-330  
    WFMInpre:XZErO, 2-330  
    WFMInpre:YMUlt, 2-331  
    WFMInpre:YOff, 2-332  
    WFMInpre:YUNit, 2-332  
    WFMInpre:YZErO, 2-333  
    WFMOutpre?, 2-333  
    WFMOutpre:BIT\_Nr, 2-334  
    WFMOutpre:BN\_Fmt, 2-334  
    WFMOutpre:BYT\_Nr, 2-335  
    WFMOutpre:BYT\_Or, 2-336  
    WFMOutpre:ENCdg, 2-336  
    WFMOutpre:NR\_Pt?, 2-337  
    WFMOutpre:PT\_Fmt?, 2-337  
    WFMOutpre:PT\_Off?, 2-338  
    WFMOutpre:PT\_ORder?, 2-338  
    WFMOutpre:WFIId?, 2-339  
    WFMOutpre:XINcr?, 2-340  
    WFMOutpre:XUNit?, 2-341  
    WFMOutpre:XZErO?, 2-341  
    WFMOutpre:YMUlt?, 2-342  
    WFMOutpre:YOff?, 2-342  
    WFMOutpre:YUNit?, 2-343  
    WFMOutpre:YZErO?, 2-343  
    WAVFrm?, 2-323

WFMInpre?, 2-324  
WFMInpre:BIT\_Nr, 2-324  
WFMInpre:BN\_Fmt, 2-325  
WFMInpre:BYT\_Nr, 2-325  
WFMInpre:BYT\_Or, 2-326  
WFMInpre:ENCdg, 2-326  
WFMInpre:NR\_Pt, 2-327  
WFMInpre:PT\_Fmt, 2-328  
WFMInpre:PT\_Off, 2-328  
WFMInpre:WFIId, 2-329  
WFMInpre:XINcr, 2-329  
WFMInpre:XUNit, 2-330  
WFMInpre:XZEro, 2-330  
WFMInpre:YMUlt, 2-331  
WFMInpre:YOFf, 2-331  
WFMInpre:YUNit, 2-332  
WFMInpre:YZEro, 2-333  
WFMOutpre?, 2-333  
WFMOutpre:BIT\_Nr, 2-334  
WFMOutpre:BN\_Fmt, 2-334  
WFMOutpre:BYT\_Nr, 2-335  
WFMOutpre:BYT\_Or, 2-335  
WFMOutpre:ENCdg, 2-336  
WFMOutpre:NR\_Pt?, 2-337  
WFMOutpre:PT\_Fmt?, 2-337  
WFMOutpre:PT\_Off?, 2-338  
WFMOutpre:PT\_ORDER?, 2-338  
WFMOutpre:WFIId?, 2-339  
WFMOutpre:XINcr?, 2-340  
WFMOutpre:XUNit?, 2-341  
WFMOutpre:XZEro?, 2-341

WFMOutpre:YMUlt?, 2-342  
WFMOutpre:YOFf?, 2-342  
WFMOutpre:YUNit?, 2-343  
WFMOutpre:YZEro?, 2-343

## Z

### Zoom commands

ZOOM:GRAticule:SIZE?, 2-344  
ZOOM:GRAticule:SPLit?, 2-344  
ZOOM:ZOOM<x>:FACtor?, 2-346  
ZOOM:ZOOM<x>:HORizontal:POSition, 2-346  
ZOOM:ZOOM<x>:HORizontal:SCAle, 2-347  
ZOOM:ZOOM<x>:POSition, 2-347  
ZOOM:ZOOM<x>:SCAle, 2-347  
ZOOM:ZOOM<x>:STATE, 2-348  
ZOOM:ZOOM<x>?, 2-345  
ZOOM?, 2-343  
ZOOM{:MODE|:STATE}, 2-345  
ZOOM:GRAticule:SIZE?, 2-344  
ZOOM:GRAticule:SPLit?, 2-344  
ZOOM:ZOOM<x>:FACtor?, 2-346  
ZOOM:ZOOM<x>:HORizontal:POSition, 2-346  
ZOOM:ZOOM<x>:HORizontal:SCAle, 2-346  
ZOOM:ZOOM<x>:POSition, 2-347  
ZOOM:ZOOM<x>:SCAle, 2-347  
ZOOM:ZOOM<x>:STATE, 2-348  
ZOOM:ZOOM<x>?, 2-345  
ZOOM?, 2-343  
ZOOM{:MODE|:STATE}, 2-345